

To the Graduate Council:

I am submitting herewith a thesis written by Ravindra V. Narkhede entitled "Fusion And Uncertainty Modelling of 3-D Multisensory Data." 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 Pace

Gene B...

Accepted for the Council:

Lawminkel

Vice Provost
and Dean of The Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at The University of Tennessee, Knoxville, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of the source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in his absence, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature 

Date May 23 1989.

FUSION AND UNCERTAINTY MODELLING OF 3-D MULTISENSORY DATA

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Ravindra V. Narkhede

August 1989

Dedication

This thesis is dedicated to my late grandparents Shripat Narkhede and Narmada Narkhede.

Acknowledgments

I wish to express my appreciation to my major advisor, Dr. M. A. Abidi, for his support, advice, and patience during the course of this research. I would also like to thank Dr. M. Pace and Dr. D. Brzakovic for serving on my committee and reviewing this thesis.

In the pursuit of this degree, I have also benefited along the way from the help of many fellow students, some now graduated . In particular, I wish to mention Dr. R. O. Eason, Vinod Shah, Hal Beck, Mishi Saravi, Chandra Tan and Hamed Sari-Sarraf. In addition, the help from Janet Smith in preparing this thesis is also very much appreciated.

Further appreciation is expressed to my parents for their support and encouragement throughout my education. Last but not the least I would like to thank my fiancée Madhuri for her constant support and patience throughout the course of my studies.

Abstract

Dealing with uncertainty for data extracted from disparate sensors is a fundamental issue in the area of sensor integration and fusion. In this thesis we describe a technique that models and fuses three dimensional information extracted from a stereo vision system and an ultrasonic range system in order to recover the three dimensional pose of an object as well as the accuracy associated with these measurement. Specifically, we have probabilistically modelled the sources of error in a stereo system and those inherent in the ultrasonic range measurement system. First the measurement generated by stereo system is updated by the ultrasonic system using an optimization technique solved for by the Euler-Lagrange Calculus of Variations equations. Second the uncertainties inherent in each of the measurements are combined by fusing the probability densities of the stereo and ultrasonic range data for each of the target points describing the pose. The advantage of this method is that we are able to deal with significantly different data in a common and unique way to infer knowledge that exploits the best features and ignores weaknesses of each sensor.

TABLE OF CONTENTS

CHAPTER	PAGE
1. Introduction	1
1.1 Related Work	3
1.2 Synopsis	11
2. Position of 3-D points from stereo	13
2.1 Introduction	13
2.2 Stereo Imaging	16
2.2.1 Application of Stereo to Robotics	19
2.2.2 Development of the Stereo Algorithm	22
2.3 Error Analysis	29
2.4 Implementation and Experimental Results	33
2.4.1 Implementation of the Algorithm	33
3. Range Sensing	41
3.1 Introduction	41
3.2 Development of the Sensor Model	42
3.3 Intensity Guided Range Sensing	49
3.3.1 Recovery of corners	49
3.3.2 Orientation of the Range Sensor	51

CHAPTER	PAGE
3.4 3-D Position from Range Data	52
3.5 Implementation and Experimental Results	55
3.6 Conclusion	60
4. Sensor Fusion	61
4.1 Joint Density Functions	61
4.1.1 p.d.f: Function of a Random Variable	62
4.1.2 p.d.f: Difference of Two Random Variables	63
4.1.3 p.d.f: Ratio of Two Random Variables	63
4.1.4 p.d.f: Product of Two Random Variables	63
4.2 Modelling of Density Functions	64
4.3 Fusion of the Density Functions	69
4.3.1 First Order Approximation of the Fusion Functional . . .	71
4.3.2 Mathematical Formulation	73
4.4 Numerical Solution	75
4.5 Implementation and Experimental Results	79
4.5.1 Synthetic Data	79
4.5.2 Real Data	86
5. Conclusion	138
BIBLIOGRAPHY	139
VITA	142

LIST OF TABLES

TABLE	PAGE
2.1 Camera Parameters	37
2.2 Object corner coordinates in pixels and <i>mm</i>	37
2.3 The 3-D coordinates of the object	39
2.4 The correspondence errors corresponding to the 3-D points . . .	39
2.5 Percent relative errors of the 3-D points in RCS	40
3.1 Range Sensor model data: the elements of the matrix represent <i>e</i> %.	48
3.2 The data set parameters	56
3.3 3-D points from range data	59
4.1 The normal density table of points obtained from stereo (dimen- sions are in millimeters)	66
4.2 3-D coordinates of the plane	133
4.3 Variance associated with the 3-D coordinates of the plane	134

LIST OF FIGURES

FIGURE	PAGE
2.1 Image Formation	15
2.2 Stereo Imaging Geometry	17
2.3 The Cincinnati Milacron 6 degrees of freedom T ³ -726 Industrial Robot	20
2.4 The Robot Coordinate System	21
2.5 The Perspective Projection Geometry of Image 1 and Image 2 . .	23
2.6 Relative camera positions.	25
2.7 Sensors mounted on the end effector of the robot	34
2.8 Image Processing Steps	36
3.1 Typical Beam Pattern (reprinted from the technical manual sup- plied by the manufacturer)	44
3.2 Range sensor with the cone	45
3.3 Range Sensor Model	46
3.4 Quadrangle with its corners and midpoints	49
3.5 Orientation Angles	52
3.6 Profile of the Range data: curve A represents the actual data set, curve B represents the ideal data set	53
3.7 Explanation of t	54
3.8 Lines in Space	55

FIGURE	PAGE
3.9 Range Data Processing	57
4.1 Density Functions for Stereo Data	67
4.2 Density Functions for Range Data	70
4.3 (a) normal input density 1 with mean = 20, (b) normal input density 2 with mean = 40. (c) Fusion function with $\lambda = 0.1$ (d) Fused density with mean = 29.94	80
4.4 For $\lambda = 0.1$: (a) Fusion function superimposed on the two densi- ties (b) Fused density superimposed on the two densities	81
4.5 Fusion for $\lambda = 0.1$: A: Input density 1. B: Input density 2. C: Fu- sion of input densities. D: The arithmetic mean, E: The geometric mean	82
4.6 (a) normal input density 1 with mean = 20, (b) normal input density 2 with mean = 40, (c) Fusion function with $\lambda = 7.5$ (d) Fused density with mean 29.1	83
4.7 For $\lambda = 7.5$: (a) Fusion function superimposed on the two densi- ties (b) Fused density superimposed on the two densities	84
4.8 Fusion for $\lambda = 7.5$: A: Input density 1, B: Input density 2. C: Fusion of A and B. D: The arithmetic mean. E: The geometric mean	85
4.9 A represents the p.d.f $p(y_1)$. B represents the p.d.f $p(y_2)$. C repre- sents the $p(y_1 - y_2)$, D represents the $p(x_1)$	87

4.10 (a): p.d.f $p_{X_1}^s$ with mean = -9.8 and variance = 11.8 . (b): p.d.f $p_{X_1}^r$ with mean = -13.5 and variance = 38.2 . (c): Fusion function with $\lambda = 0.1$ (d): Fused density with mean -10.8 and variance 23.91	89
4.11 For $\lambda = 0.1$: (a) Fusion function superimposed on the two densities (b) the fused density superimposed on the two densities . .	90
4.12 Fusion for $\lambda = 0.1$: A: Input density 1, B: Input density 2, C: Fusion of the two inputs, D: Arithmetic mean, E: Geometric mean	91
4.13 (a) p.d.f $p_{X_1}^s$ with mean = -9.8 and variance = 11.8 , (b): p.d.f $p_{X_1}^r$ with mean = -13.5 and variance = 38.2 , (c) Fusion function with $\lambda = 0.25$ (d) Fused density with mean -11.23 and variance 23.9	93
4.14 For $\lambda = 0.25$: (a) Fusion function superimposed on the two densities (b) the fused density superimposed on the two densities . .	94
4.15 Fusion for $\lambda = 0.25$: A: Input density 1, B: Input density 2, C: Fusion of the two inputs, D: Arithmetic mean, E: Geometric mean	95
4.16 (a) p.d.f $p_{X_1}^s$ with mean = -9.8 and variance = 11.8 , (b): p.d.f $p_{X_1}^r$ with mean = -13.5 and variance = 38.2 , (c) Fusion function with $\lambda = 0.5$ (d) Fused density with mean -11.32 and variance 26.6	96
4.17 For $\lambda = 0.5$: (a) Fusion function superimposed on the two densities (b) the fused density superimposed on the two densities . . .	97

4.18	Fusion for $\lambda = 0.5$: A: Input density 1, B: Input density 2, C: Fusion of the two inputs, D: Arithmetic mean, E: Geometric mean	98
4.19	(a) p.d.f $p_{X_1}^s$ with mean = -9.8 and variance = 11.8 , (b): p.d.f $p_{X_1}^r$ with mean = -13.5 and variance = 38.2 . (c) Fusion function with $\lambda = 0.75$ (d) Fused density with mean -10.9 and variance 22.9	99
4.20	For $\lambda = 0.75$: (a) Fusion function superimposed on the two den- sities (b) the fused density superimposed on the two densities	100
4.21	Fusion for $\lambda = 0.75$: A: Input density 1, B: Input density 2, C: Fusion of the two inputs, D: Arithmetic mean, E: Geometric mean	101
4.22	(a) p.d.f $p_{X_1}^s$ with mean = -9.8 and variance = 11.8 , (b): p.d.f $p_{X_1}^r$ with mean = -13.5 and variance = 38.2 , (c) Fusion function with $\lambda = 2.0$ (d) Fused density with mean -11.0 and variance 25.9	102
4.23	For $\lambda = 2.0$: (a) Fusion function superimposed on the two den- sities (b) the fused density superimposed on the two densities	103
4.24	Fusion for $\lambda = 2.0$: A: Input density 1, B: Input density 2, C: Fusion of the two inputs, D: Arithmetic mean, E: Geometric mean	104
4.25	(a) p.d.f $p_{X_1}^s$ with mean = -9.8 and variance = 11.8 , (b): p.d.f $p_{X_1}^r$ with mean = -13.5 and variance = 38.2 . (c) Fusion function with $\lambda = 2.5$ (d) Fused density with mean -11.96 and variance 29.6	105

4.26	For $\lambda = 2.5$: (a) Fusion function superimposed on the two densities (b) the fused density superimposed on the two densities . . .	106
4.27	Fusion for $\lambda = 2.5$: A: Input density 1, B: Input density 2, C: Fusion of the two inputs, D: Arithmetic mean, E: Geometric mean	107
4.28	(a) p.d.f $p_{X_1}^s$ with mean = -9.8 and variance = 11.8 , (b): p.d.f $p_{X_1}^r$ with mean = -13.5 and variance = 38.2 , (c) Fusion function with $\lambda = 5.0$ (d) Fused density with mean -12.3 and variance 30.4108	
4.29	For $\lambda = 5.0$: (a) Fusion function superimposed on the two densities (b) the fused density superimposed on the two densities . . .	109
4.30	Fusion for $\lambda = 5.0$: A: Input density 1, B: Input density 2, C: Fusion of the two inputs, D: Arithmetic mean, E: Geometric mean	110
4.31	(a) p.d.f $p_{X_1}^s$ with mean = -9.8 and variance = 11.8 , (b): p.d.f $p_{X_1}^r$ with mean = -13.5 and variance = 38.2 , (c) Fusion function with $\lambda = 7.5$ (d) Fused density with mean -12.3 and variance 30.4111	
4.32	For $\lambda = 7.5$: (a) Fusion function superimposed on the two densities (b) the fused density superimposed on the two densities . . .	112
4.33	Fusion for $\lambda = 7.5$: A: Input density 1, B: Input density 2, C: Fusion of the two inputs, D: Arithmetic mean, E: Geometric mean	113
4.34	(a) p.d.f $p_{X_1}^s$ with mean = -9.8 and variance = 11.8 , (b): p.d.f $p_{X_1}^r$ with mean = -13.5 and variance = 38.2 , (c) Fusion function with $\lambda = 10.0$ (d) Fused density with mean -12.24 and variance 30.3	114

4.35	For $\lambda = 10.0$: (a) Fusion function superimposed on the two densities (b) the fused density superimposed on the two densities . .	115
4.36	Fusion for $\lambda = 10.0$: A: Input density 1. B: Input density 2. C: Fusion of the two inputs. D: Arithmetic mean of E: Geometric mean of A	116
4.37	(a) p.d.f $p_{X_1}^s$ with mean = -9.8 and variance = 11.8 . (b): p.d.f $p_{X_1}^r$ with mean = -13.5 and variance = 38.2 . (c) Fusion function with $\lambda = 1.0$ (d) Fused density with mean -10.4 and variance 20.9	118
4.38	For $\lambda = 1.0$: (a) Fusion function superimposed on the two densities (b) the fused density superimposed on the two densities . . .	119
4.39	Fusion for $\lambda = 1.0$: A: Input density 1, B: Input density 2, C: Fusion of the two inputs, D: Arithmetic mean, E: Geometric mean	120
4.40	(a) p.d.f $p_{Y_1}^s$ with mean = 3.9 and variance = 2.5 , (b) p.d.f $p_{Y_1}^r$ with mean = 0.3 and variance = 34.2 (c) Fusion function with $\lambda = 2.0$ (d) Fused density with mean 1.6 and variance 23.6 . . .	122
4.41	(a) p.d.f $p_{Z_1}^s$ with mean = 31.9 and variance = 61.8 . (b) p.d.f $p_{Z_1}^r$ with mean = 34.8 and variance = 76.0 . (c) Fusion function with $\lambda = 5.0$ (d) Fused density with mean 33.8 and variance 67.1 . . .	123
4.42	(a) p.d.f $p_{X_2}^s$ with mean = -4.3 and variance = 1.9 , (b) p.d.f $p_{X_2}^r$ with mean = -8.8 and variance = 101.8 . (c) Fusion function with $\lambda = 2.5$ (d) Fused density with mean -7.5 and variance 54.93 . .	124

4.43	(a) $p_{Y_2}^s$, (b) $p_{Y_2}^r$, (c) Fusion function with $\lambda = 5.0$ (d) Fused density with mean -7.1 and variance 0.2	125
4.44	(a) p.d.f $p_{Z_2}^s$ with mean $= 26.9$ and variance $= 44.3$, (b) p.d.f $p_{Z_2}^r$ with mean $= 23.8$ and variance $= 166.1$. (c) Fusion function with $\lambda = 5.0$ (d) Fused density with mean 30.1 and variance 89.5 . . .	126
4.45	(a) p.d.f $p_{X_3}^s$ with mean $= 0.7$ and variance $= 0.6$, (b) p.d.f $p_{X_3}^r$ with mean $= -3.3$ and variance $= 36.3$. (c) Fusion function with $\lambda = 1.0$ (d) Fused density with mean -1.75 and variance 26.2 . .	127
4.46	(a) $p_{Y_3}^s$, (b) $p_{Y_3}^r$, (c) Fusion function with $\lambda = 5.0$ (d) Fused density with mean -7.9 and variance 0.7	128
4.47	(a) p.d.f $p_{Z_3}^s$ with mean $= 31.9$ and variance $= 61.85$, (b) p.d.f $p_{Z_3}^r$ with mean $= 34.8$ and variance $= 78.9$, (c) Fusion function with $\lambda = 5.0$ (d) Fused density with mean 34.2 and variance 68.9	129
4.48	(a) p.d.f $p_{X_4}^s$ with mean $= -0.46$ and variance $= 0.6$, (b) p.d.f $p_{X_4}^r$ with mean $= -4.1$ and variance $= 27.8$, (c) Fusion function with $\lambda = 1.0$ (d) Fused density with mean -2.2 and variance 17.3	130
4.49	(a) p.d.f $p_{Y_4}^s$ with mean $= 4.6$ and variance $= 2.9$, (b) p.d.f $p_{Y_4}^r$ with mean $= 0.8$ and variance $= 29.6$. (c) Fusion function with $\lambda = 1.0$ (d) Fused density with mean 3.9 and variance 9.5	131
4.50	(a) p.d.f $p_{Z_4}^s$ with mean $= 31.9$ and variance $= 61.9$. (b) p.d.f $p_{Z_4}^r$ with mean $= 34.9$ and variance $= 72.4$. (c) Fusion function with $\lambda = 5.0$ (d) Fused density with mean 33.9 and variance 64.8 . . .	132

FIGURE

PAGE

4.51 (a) Graphical representation of the variance associated to the stereo data. (b) Graphical representation of the variance associated to the range data	136
4.52 Variance in stereo, range data: the dotted curve represents the expected variance after fusing	136

CHAPTER 1

Introduction

In recent years an increased interest in the development of multi-sensor robotic systems is seen [1,2,3,4]. The reason for this can be attributed to the fact that there is a fundamental limitation on the use of a single sensor to reconstruct the description of an environment and because of the large increase in the performance/cost ratio of these sensors. For a robotic system to achieve a degree of intelligence and autonomy, it must be capable of using different sources of sensory information in an active (real time processing of data) and dynamic (adapting to changes in the environment) manner.

Acquiring data from multiple sensors enables a robot involved in inspection and manipulation tasks to operate accurately and efficiently. Robust task execution is accomplished by increasing sensor and manipulator accuracy, using high tolerance and known objects, and supplying parts in predetermined pose. This works well when the robot controlled environment is known and all the tasks and motions to be performed are planned in advance. If robotic systems are increasingly required to operate in complex domains, where the environment and the tasks are unknown or uncertain, then this allowance for uncertainty will be insufficient to ensure safe, efficient, and accurate operation of the robotic

system.

An additional advantage in using several different sensors is that, different sensors are sensitive to different signals, each of which unveils a particular set of properties of the sensed environment. For example, intensity images are rather easily and cheaply acquired. However, extracting three-dimensional information from those is often a difficult task and requires sufficient levels of illumination to be effective. On the contrary, range imaging though costly and cumbersome, contains explicit three-dimensional and shape information.

A general robot vision or scene analysis system should be capable of locating the objects in a scene and interpreting the three-dimensional (3-D) relationships among them. In most cases, the scene is represented by an image or a set of images taken from different locations and viewing angles. A scene analysis module is required to locate the objects and interpret their mutual relationships from a number of given images. Locating the objects in a two-dimensional (2-D) image may be considered as low level analysis, while interpreting the geometrical structures or the mutual relationships of the located objects may be considered as high level analysis. Although this distinction of the two different levels is not particularly clear, the low-level analysis is often concerned with feature extraction such as image segmentation, and the high-level analysis with interpreting the image data in three dimensions in terms of the segmented features and the mutual relationships of these features.

The purpose of this thesis is to formulate a method to fuse data from dis-

parate sensors to get a sensor independent "picture" of the environment. Stereo imaging is used to locate a 3-D object (a quadrangular surface) and the range sensor is guided using the intensity data to get the range information of the object. In the following we review a number of existing studies towards vision, range, error analysis, and multi-sensor integration.

1.1 Related Work

Surfaces, edges, and vertices are some of the features used to describe the spatial discontinuities of an object in a 3-D world. These features characterize the shape of an object in space and thus are appropriate for use in scene description. Most of the early work in object location dealt with interpretation of 3-D objects from a single view. The object in the image is represented by a wire frame model in which vertex types are classified. The object is then located according to certain vertex connection properties [5]. In the past, the development of the high-level object interpretation process was almost independent of the low-level feature extraction process. To extract image features such as occluding edges, concave edges, or convex edges uniquely is very difficult unless specific semantic description is imposed or global information is used. Hung *et al.* [6] proposed a procedure which utilizes knowledge about the dimensions (*i.e.*, the distance between each of the six pairs of the four vertices) of a 3-D planar quadrangle to determine its position and orientation with respect to the camera-centered

coordinate system. Based on this procedure, they then presented a method to determine the pose of the camera with respect to a fixed world coordinate system, if the coordinates of the vertices of the quadrangle in the world frame are also available.

When no prior information concerning the robots environment is available, an approach for object location is to use depth information. Depth information for an object may be obtained in two different ways. One is to use a pair of stereo images [7,8,9], and the other is to use a range sensor to extract the range information [10,11] to the object. Using the range information, the description of an object is carried out, not on the basis of a single 2-D projection of the 3-D object, but on the basis of 3-D (2-D image and depth) information about the object. Usually, a computer-controlled high precision sensor system is required for collecting the range information [10,11] which can be a very lengthy and time consuming process. Although, today's range sensors can be very efficient in terms of the time taken to acquire a range image, their cost is prohibitive. A range sensor is an active device which may not be suitable for certain constrained environments. Use of stereo is attractive in that it needs only pictorial information to obtain the 3-D information about the object.

A stereo pair of images can be obtained either from two cameras or from one camera occupying two different viewing geometries. A significant advantage of this approach is that it is passive, relying solely on object illumination. However, to locate a 3-D object, corresponding image points in the two given images

must be found. Most of the work in using stereo for object location has been devoted to solving the ever challenging correspondence problem [7,8,9,12]. The triangulation method [13,14,15,16] is widely used in computing the location of a 3-D object from a pair of the corresponding points in two images. A point in one image determines a ray in space; the corresponding point in the second image determines another ray corresponding to the same object point. Thus, knowing the model of the camera, the 3-D object location may be computed using trigonometric relations.

The accuracy of object location depends directly on the angular distance between the two cameras. Using wide angle stereo images in object location usually results in more accurate depth information [9] because the disparities between the two images increases proportionately to the distance between the two cameras. The increase in the disparities complicates the problem of searching for the corresponding points. A traditional method for locating the corresponding points is template correlation [7,8,9]. A template constructed with certain adjacent neighbor points to a reference point in one image is used in a search for the matching point in the other image based on crosscorrelation. Usually, a template with fixed size and shape is used for this purpose. However, information in the template region may be quite different in the two images because of changes in perspective and changes in surface reflectivity with the imaging angles. Another method for searching for corresponding points is to match features from the two images. Recent work in 3-D object location using features can be found in [12].

Feature-based matching is much more efficient since only a few correspondences need to be determined to locate the object in a 3-D world.

If a point in a scene has been correctly identified in each image then its 3-D position can be recovered. The relationship between the geometry of the stereo camera set-up and the accuracy in obtaining the actual 3-D positions has received scant attention though it is of great practical importance. Duda and Hart [16] give a brief treatment of the subject. To obtain a useful 3-D description of a scene, the absolute error after computing this 3-D point has to be minimum for a robust robotic system. Verri *et al.* [17] showed that, as expected, the absolute error in *depth* becomes smaller as the difference between the points of view of the two images becomes larger. Using some constraints on the mechanical type of errors they resolve the total error down to only the correspondence error in the two images. They show that the precision of the setup geometry, affects the absolute depth estimates, while the mechanical type of errors affect the relative depth estimates. Huang's [18] paper deals mainly with developing all the partial derivatives which are the error transmission functions. This study is directed towards determining mechanical errors. McVey and Lee [19] have performed a worst case error analysis on the image plane resolution required to achieve depth measurement of a given accuracy. As opposed to the worst case error analysis and ignoring camera lens distortion and other optical nonlinearities, Bolstein *et al.* [20] determine the probability that a certain position estimate is within a specified tolerance given the camera geometry of the stereo set-up.

Recovering depth from stereo is an indirect method and is also a heavy computational burden as establishing the corresponding points in the two images is essentially a search problem. On the other hand, several studies have used a direct approach in measuring the range which involves producing an energy beam, directing it on the scene, and measuring the reflected signal. A significant amount of current literature in robot sensing is devoted to range sensing [21]. Boyter [22] describes an approach for matching the 3-D primitives extracted from a range image with 3-D model descriptions in a database. Vemuri and Aggarwal [23] present a technique for determining the orientation and identity of an object based on matching object and model surface descriptions. Partial information about the object is provided in the form of range data acquired from a single view. The objects and models are represented by regions that are a collection of surface patches homogeneous in curvature-based properties. This technique for determining the orientation requires that correspondence be established exactly, between one point on the object surface and one on the model surface.

Unfortunately, the time required to fully sense a range image is long relative to the time required to sense an intensity image. Conversely, a single intensity image lacks the depth information required to construct 3-D object descriptors. In recent years an increased interest in the development of multi-sensor robotic systems is seen [1,2,3,4].

Tenenbaum [1] reports the preliminary work done on a knowledge-based

perceptual system for a robot that must function in an actual office environment. This system relies on intensity (color) and range data to increase the likelihood of finding distinguishing surface attributes for a particular object. Descriptive representation for complex attributes (*e.g.*, shape and texture) are avoided in favor of simple representation sufficient to distinguish the object of interest. This approach was further carried out by Garvey [2] whose goal oriented perception system locates objects in multisensory images, by planning and executing task oriented strategies. These strategies use various kinds of knowledge including object descriptions, world models, and sensor models to determine the features that distinguish the target object from other known objects. This strategy which is based on the optimization of the likelihood of success of the search and the minimization of the operation time allows the system to sense only the data necessary to complete any particular step. In doing so, it best addresses the problem of sensory data overload. Graham [4] provided an overview of some of the issues involved in fusing information. He described a specific fusion model developed for a multisensory collision avoidance system. This system combined “qualitative” sensory information using discretized world model. This was subjected to probability of the sensors ability to detect; a correct judgement of the “safe move” and the “unsafe move” hypothesis, and four (ultrasound, infrared, capacitive, and microwave) sensor model.

Magee *et al.* [24] presented a method for recognizing 3-D objects in multisensory images. The method utilizes both intensity and range data for recognition.

It avoids the time consuming task of sensing and processing an entire range image. Their approach consists of recording an intensity image, extracting relevant features from it, then selecting the critical areas for which range information is needed. Depth is then measured for these preselected regions. Aggarwal and Magee [25] pursued this approach of intensity guided range sensing in order to determine motion parameters.

Duda *et al.* [10] described a laser range finder measuring the phase difference between the emitted wave-modulated laser beam and the reflected wave as well as the amplitude of the reflected wave. These two measurements yielded registered range and brightness images of the scene. They have shown how to use the data in order to perform low level tasks such as the detection of jump boundaries, extraction of planar surfaces, and the determination of the normal view of slanted planar surfaces. The last task involves combining range and intensity data in two distinct steps. First, range data is utilized in order to obtain the location and orientation of an oblique plane. Then, after this plane has been uniformly sampled in its own 2-D cartesian coordinate system, the intensity value is obtained at these points by interpolating among the recorded intensity values. The perspective distortion is thus corrected [11].

Gil *et al.* [26] addressed the issue of image registration between intensity and range edge maps to combine them in a unique form for segmentation. In their study, the Kirsh operator was selected for the computation of the intensity gradient, and the range edges were obtained by means of the evaluation of

the surface curvature at every pixel location. Two approaches are proposed for the edge map combination. The first one consists of extracting edges appearing in both edge maps by means of a local AND operator. The pixel values from the intensity edge map are AND-ed with the $n \times n$ neighborhood of the corresponding intensity edge map extracted from the range image. The second one is more intricate. It also consists of extracting common edges, but this task is based on a global AND operation. Entire edges are extracted from the intensity image and traced in the range edge map, and filled. This approach allows for the elimination of spurious short edges by also setting a minimal edge length.

Harmon *et al.* [27] proposed a “distributed blackboard”. Processing is distributed among three subsystems sensor, control, and knowledge base. Each subsystem has its own world model which is organized on a blackboard that resides in the subsystems shared memory. The blackboard is organized as a classtree data structure. In this fashion, objects can be related to one another as well as to the data being processed. Data from different spatial locations or different time lapses can be incorporated and maintained on the blackboard and referred to as needed. Error estimation, confidence factors, and object relationships are the parameters maintained in the blackboard. They are used to fuse data from multiple sensors.

Kent *et al.* [28] used multiple sources of information to construct an internal representation of the robot workspace. They begin by dividing their representation scheme into two levels. The first level keeps track of the actual workspace,

while the second keeps track of the objects and their features which make up the workspace. The spatial representation is organized as an octree and the feature-based representation is linked to both the octree and the generic object models (CAD descriptors stored in a knowledge base). These models are rotated and translated into the actual real world position as defined by the sensory data.

Delcroix [29] presented a new method for the fusion of range and intensity edge maps. This method focuses on the integration of registered information in order to increase the confidence in the presence/absence of edges in a scene. This work is based on two constraints: the principal of existence tends to maximize the value of the output edge map at a given location if one edge map features a strong edge, and the principle of confirmability, adjusts this value according to the edge content in the other edge map at the same location by maximizing the similarity between them. The maximizations are achieved using the Euler-Lagrange equations. This method is extended to color edge maps.

1.2 Synopsis

The remainder of this thesis is organized as follows. Chapter 2 presents the analytical formulation of the recovery of depth information using a stereo algorithm. Analytical formulation of error analysis is presented for the coordinates of a number of 3-D points. Experimental results are tabulated. Chapter 3 presents the range sensor model and its formulation. An intensity guided range

data acquisition of the features extracted from the intensity image is presented. Experimental results are also tabulated. Chapter 4 models the two sensory data as probability density functions and fuses them. Conclusions are given in Chapter 5.

CHAPTER 2

Position of 3-D points from stereo

This chapter reviews the image formation process, discusses the projection of 3-D world point onto the image plane, points-out the missing depth information in a single image, then shows how *stereo* vision is used to recover this missing information. Section 2.2 illustrates the basic geometry of stereo with a simple example. Subsections 2.2.1 and 2.2.2 illustrate stereo vision applied in robotic tasks. Section 2.3 finds the *absolute relative error* of the 3-D points. Section 2.4 implements the stereo algorithm and tabulates the results for a quadrangular object.

2.1 Introduction

An image is a two-dimensional pattern of brightness. In order to analyze it we must first know how it is formed. Often, cameras are modeled by a pinhole system. The image formation process is modeled by a *perspective* projection. The projection of a point P (world point) is defined by the intersection of a projection plane (image plane), with a projection ray (light ray) emanating from the center of the projection and passing through P . The other type of projection is the *orthographic* projection, and is applied to image formation

when the distance between the projection plane and the center of the projection is very large. As an example of *orthographic* projection modeling, consider the case of aerial photography, wherein the size of the objects is small compared to the distance between the objects and the image plane.

A perspective projection model of the image formation process is shown in Fig. 2.1. A right-hand coordinate system is introduced with its origin at the pinhole. The Z-axis is aligned with the optical axis of the camera lens and points towards the image plane. The optical axis is assumed to be perpendicular to the image plane. The image plane lies at a distance λ from the pinhole, while x' and y' are the coordinates of the point p' in the image plane.

The objective is to compute where the projection p' of P on some object in front of the camera will appear on the image plane, *i.e.*, the relationship that gives the coordinates (x', y') of the projection of the point $P(X, Y, Z)$ onto the image plane. This is easily accomplished through similar triangles.

$$\frac{x'}{\lambda} = \frac{X}{Z}$$

$$\frac{y'}{\lambda} = \frac{Y}{Z}$$

or equivalently

$$X = \frac{x'}{\lambda} Z \quad (2.1)$$

$$Y = \frac{y'}{\lambda} Z \quad (2.2)$$

Equations 2.1 and 2.2 show that, unless additional information is available as to which 3-D point generated a given image point (for example, its

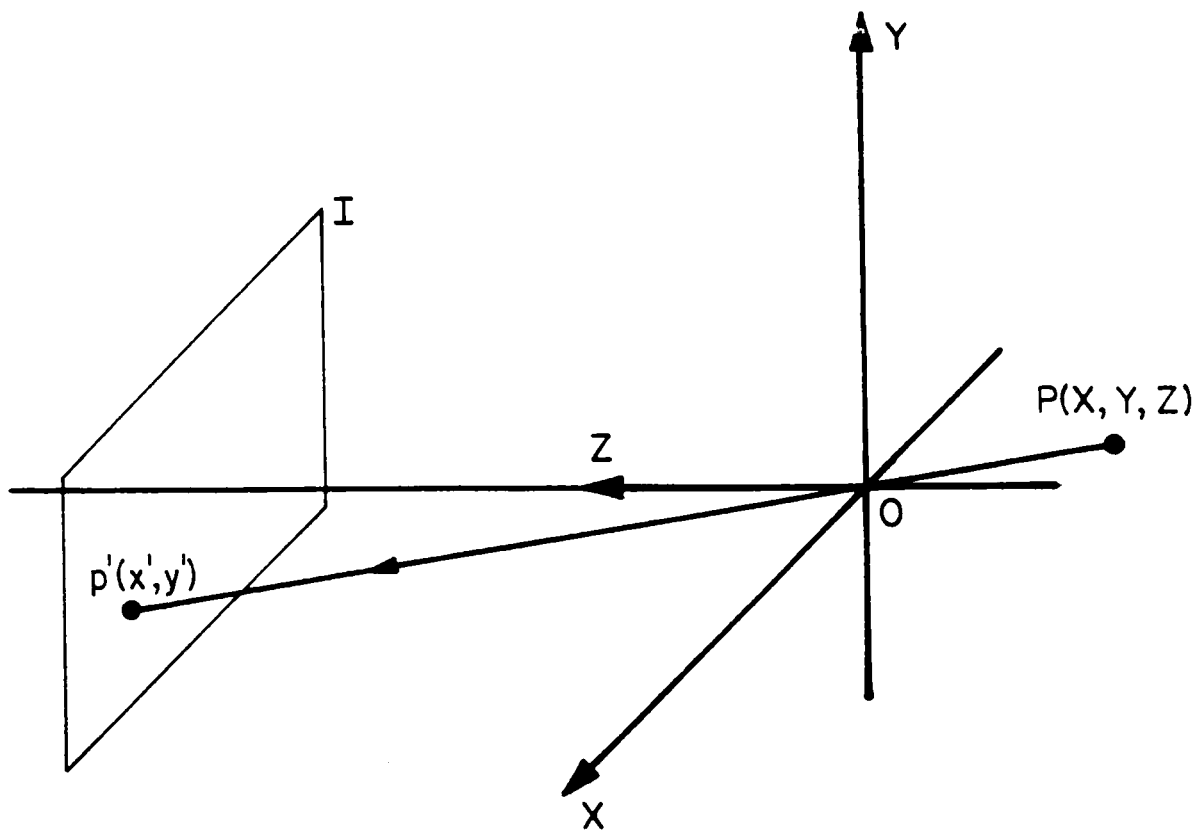


Figure 2.1: Image Formation: I : image plane, P : world point, p' : projection of P

Z-coordinate), we cannot completely recover the 3-D point from its image. This process of recovering the 3-D point from its image is known as the inverse perspective transformation [14].

2.2 Stereo Imaging

In the previous section it was shown that an image point does not uniquely determine the location of a corresponding world point. This section shows how this missing information of *depth* can be recovered using *stereoscopic* (stereo) imaging techniques. The basic camera geometry for stereo photography is illustrated in Fig. 2.2.

Two cameras are attached such that their optical axes are parallel to each other and are separated by a distance b . It is also assumed that the *baseline* is perpendicular to the optical axes and that the x -axes (X_l and X_r) are oriented to be parallel to the baseline, *i.e.*, both coordinate systems are coincident. The coordinates of the point P , $(X, Y, Z)^t$, in the world are measured relative to an origin midway between the two optical axes. The image coordinates in the left and right images are denoted by (x_l, y_l) and (x_r, y_r) , respectively.

Relative to Fig. 2.2

$$\begin{aligned}\frac{x_l}{\lambda} &= \frac{X + b/2}{Z} \\ \frac{x_r}{\lambda} &= \frac{X - b/2}{Z} \\ \frac{y_l}{\lambda} &= \frac{y_r}{\lambda} = \frac{Y}{Z},\end{aligned}$$

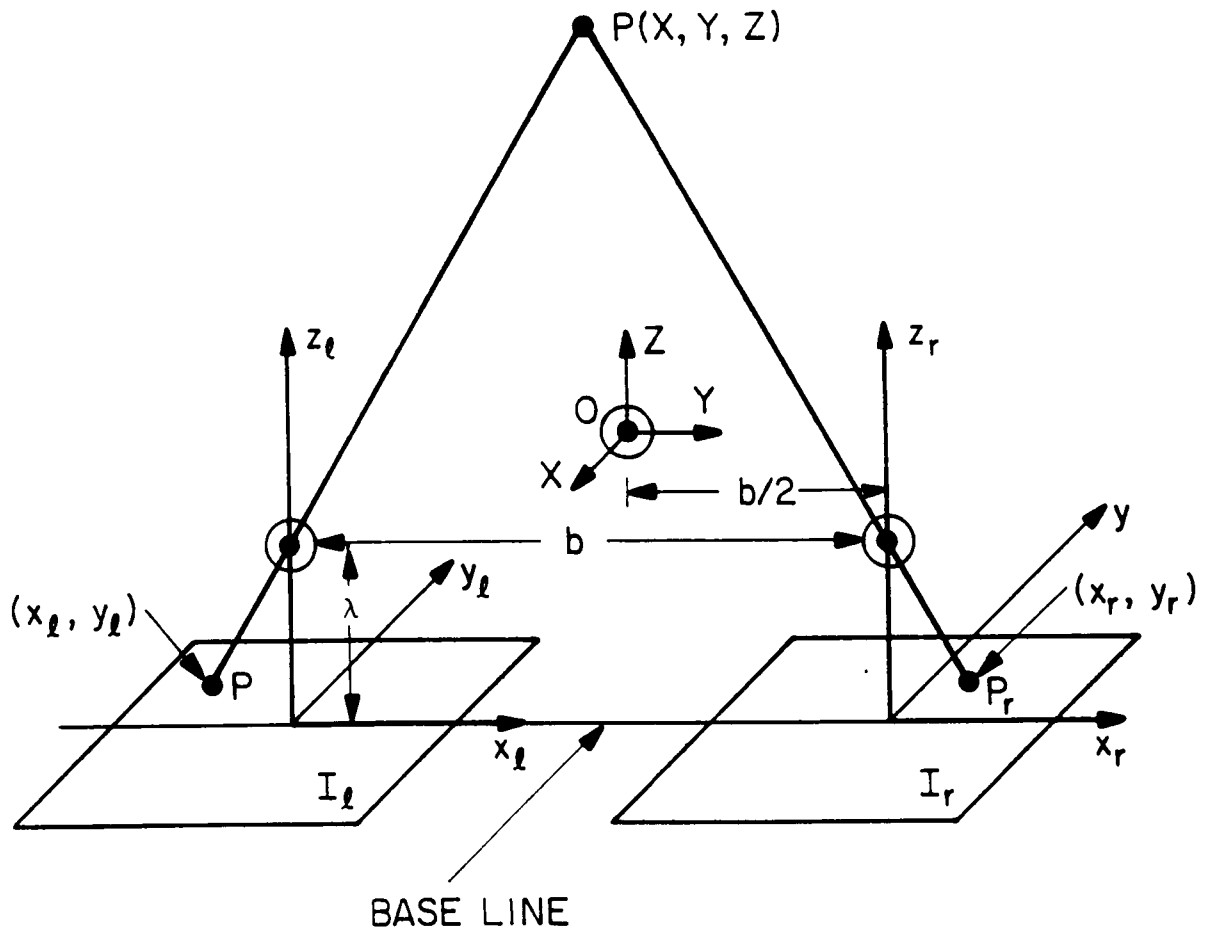


Figure 2.2: Stereo Imaging Geometry

where λ is the distance from the lens center to the image plane in both the cameras. This yields

$$X = b \frac{(x_l + x_r)/2}{x_l - x_r}, \quad (2.3)$$

$$Y = b \frac{(y_l + y_r)/2}{x_l - x_r}, \quad (2.4)$$

$$Z = b \frac{\lambda}{x_l - x_r}. \quad (2.5)$$

The distance $d = x_l - x_r$ is called the *disparity* in image coordinates. As seen from Eq. 2.5, depth is directly proportional to b and inversely proportional to the disparity [15]. Thus, *depth* information of a world point can be recovered from its projection on the image plane under some restrictions. Even if these restrictions are met, the most difficult task in stereoscopic imaging is to find two corresponding points in different images of the same scene. To establish the fact that p_l and p_r are the projections of the same point P onto I_l and I_r is referred to as solving the *correspondence problem*.

One of the two common approaches used to solve the correspondence problem is to select a point within a small region in one of the views, and attempt to find the best matching region in the other view by using correlation-based techniques. The other approach is feature-based: it is used when the scene contains distinct features such as prominent corners. In this thesis, we chose the later approach to solve this problem since the object under consideration is a quadrangular plane which has well defined edges and corners.

2.2.1 Application of Stereo to Robotics

In the previous section we discussed how depth information of a 3-D point can be recovered from two images. We assumed that the 3-D point coordinate system was coincident with the two camera coordinate systems then displaced by $b/2$ from both these systems. In our robotic system, since only one camera is mounted on the end-effector, the problem of characteristic matching is not an issue. In this section we develop an "algorithm" to find the position of a 3-D point in the real world (robots environment).

Figure 2.3 shows the robot used in our experiments. We begin by fixing a coordinate system $(X, Y, Z)^T$ in the center of the robot body [Fig. 2.3], henceforth, this coordinate system will be referred to as the robot coordinates system (RCS). The objective is to find the position $(X, Y, Z)^T$ of a 3-D point in this coordinate system. The RCS is chosen such that it obeys the right-hand rule. The navigation angles are measured anticlockwise for positive direction in the Euler notation. The Euler angles α , β , and θ representation correspond to the following sequence of rotations:

1. A rotation of α about the Z -axis
2. A rotation of β about the rotated Y^* -axis
3. A rotation of θ about the rotated Z -axis

This is illustrated in Fig. 2.4. The camera mounted on the end-effector of the robot is defined within the RCS.

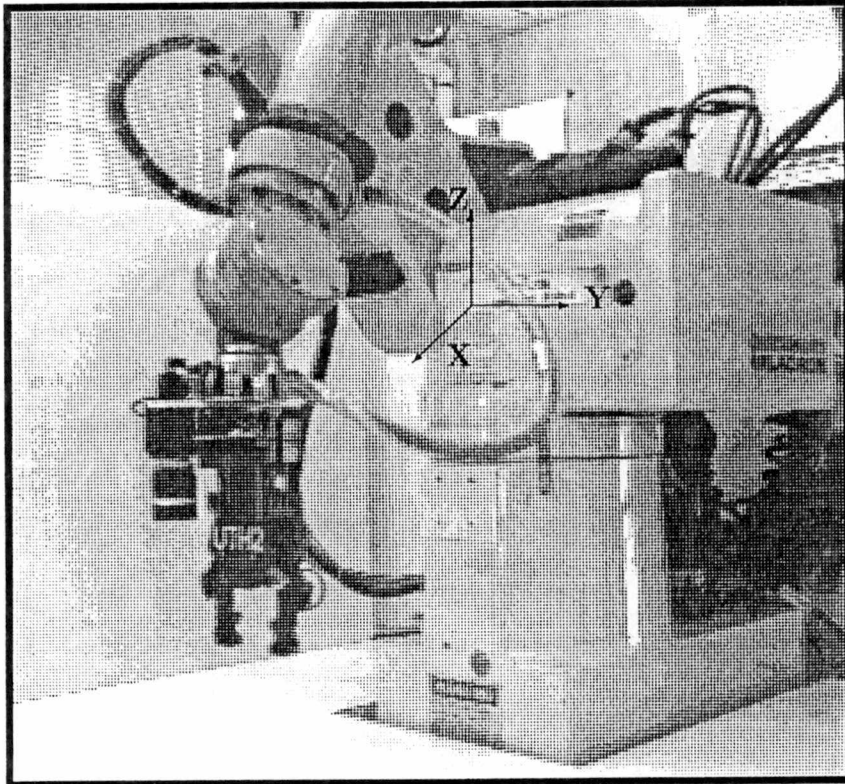


Figure 2.3: The Cincinnati Milacron 6 degrees of freedom T³-726 Industrial Robot

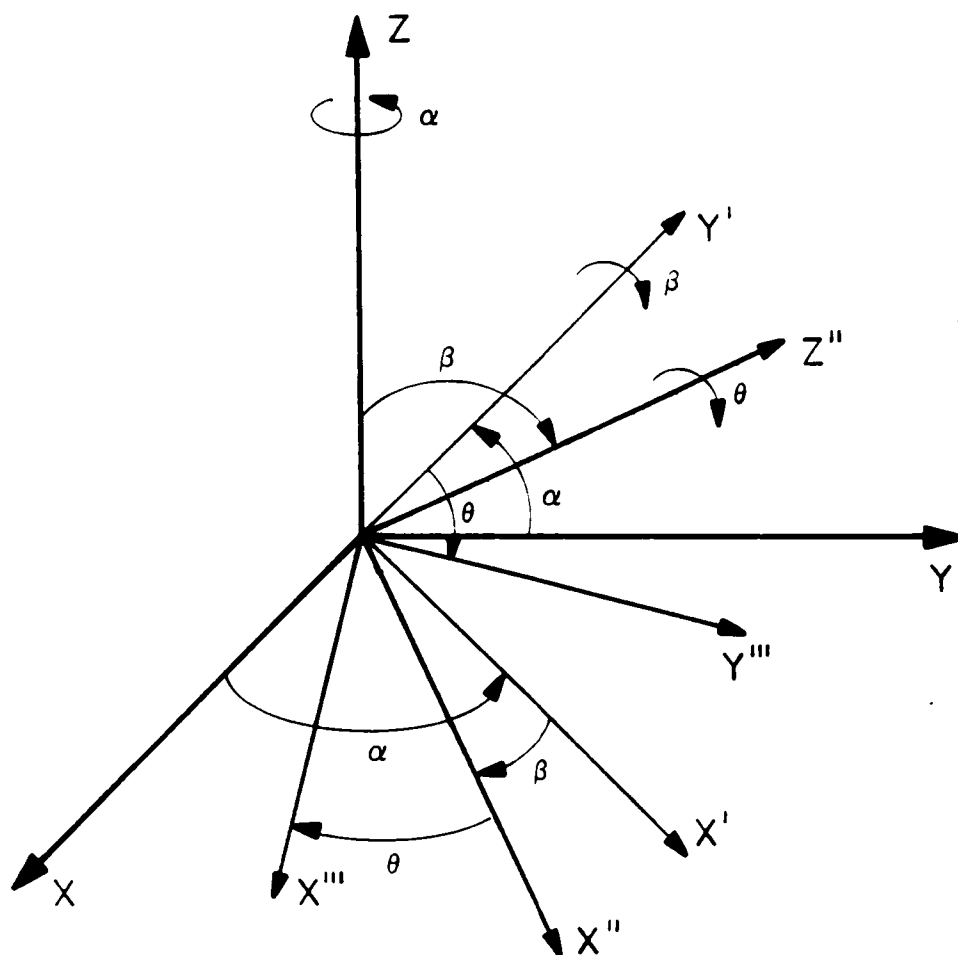


Figure 2.4: The Robot Coordinate System: α rotation about the Z -axis, β rotation about the Y' -axis, θ rotation about the Z'' -axis

2.2.2 Development of the Stereo Algorithm

The robot camera is positioned such that the object is in the camera view. This point is denoted as C_p^1 (read as camera position 1). The translation of the camera to C_p^1 is noted as $(s_1, s_2, s_3)^T$. This point $C_p^1(s_1, s_2, s_3)^T$ henceforth will be referred to as the *basepoint*. The line joining the center of the lens and the basepoint is called the Z_1 -axis and is oriented such that the direction from the basepoint to the lens is the positive direction. In order to fix a coordinate system at the base point as the origin, two other lines are chosen such that they along with the Z_1 -axis are mutually orthogonal. These are the X_1 -axis and the Y_1 -axis, their directions chosen freely. Thus a $X_1Y_1Z_1$ coordinate system is formed at C_p^1 .

In $X_1Y_1Z_1$ system the lens is at $(0, 0, \lambda)$, where, λ is the focal length of the camera. Consider a point P in the 3-D space (within the cameras viewing angle). Assume that its $X_1Y_1Z_1$ -coordinates are $(X_1, Y_1, Z_1)^T$. The line joining P to $(0, 0, \lambda)^T$ will meet the X_1Y_1 -plane (image plane) at some point. Assume its (X_1, Y_1, Z_1) coordinates to be $(x_1^p, y_1^p, 0)^T$. Then, from the geometry shown in Fig. 2.5

$$X_1 = x_1^p \frac{(\lambda - Z_1)}{\lambda}, \quad (2.6)$$

$$Y_1 = y_1^p \frac{(\lambda - Z_1)}{\lambda}. \quad (2.7)$$

In order to obtain a stereo image of the same scene the camera is now moved to another point with reference to the $X_1Y_1Z_1$ -system and whose coordinates

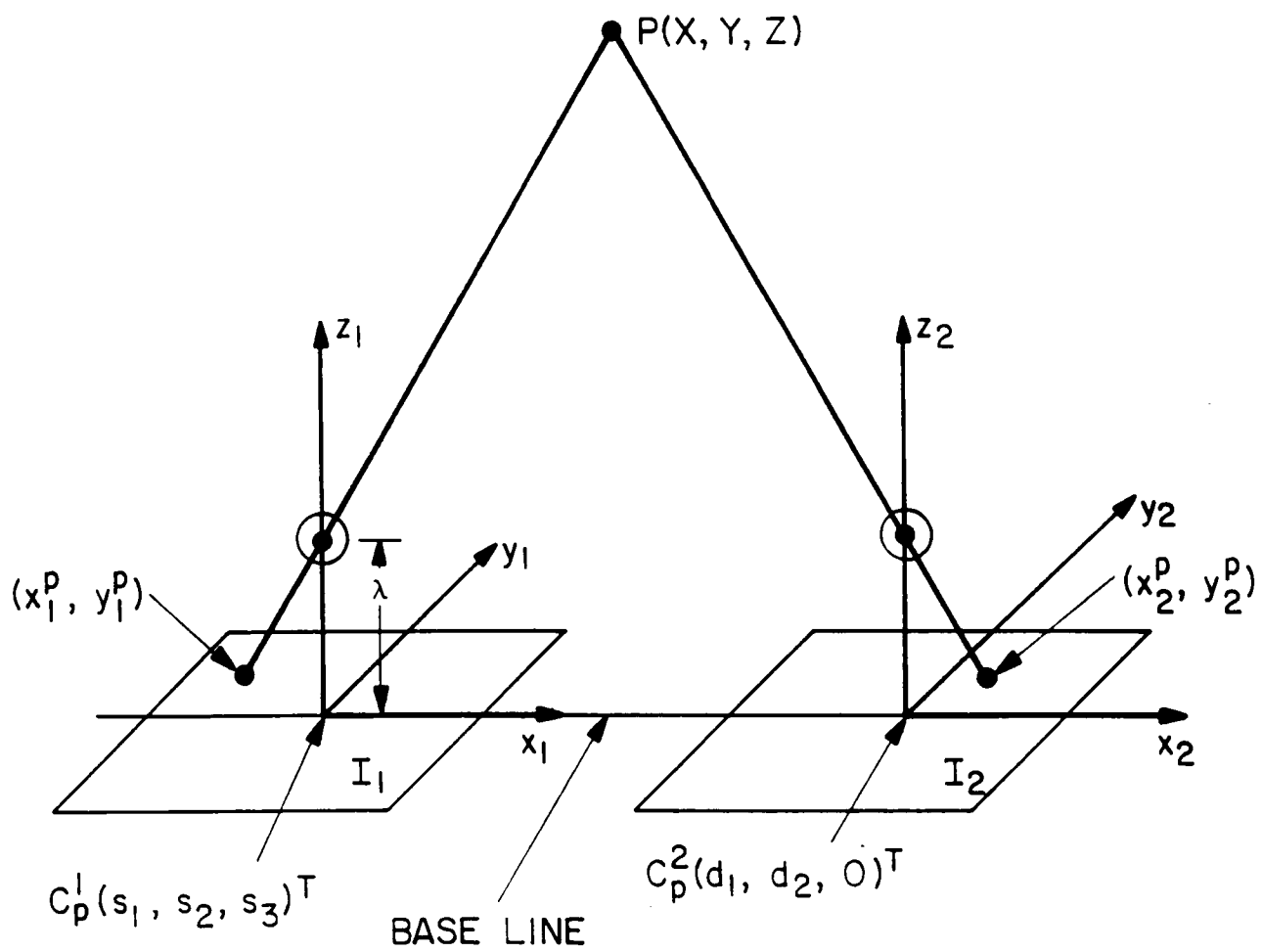


Figure 2.5: The Perspective Projection Geometry of Image 1 and Image 2

with respect to this system are $C_p^2(d_1, d_2, 0)$. Choose C_p^2 (read as camera position 2) such that the line joining the center of the lens is parallel to the Z_1 -axis and is positive in the same direction. This line is called the Z_2 -axis. Let X_2, Y_2 be the two unique lines passing through C_p^2 and parallel to X_1, Y_1 -axes, respectively. The positive directions of X_2, Y_2 axes are the same as X_1, Y_1 axes. These two lines X_2, Y_2 form the X_2, Y_2 -axes of this new coordinate system, respectively. Thus a $X_2Y_2Z_2$ coordinate system is formed at C_p^2 . The lens is again at $(0, 0, \lambda)^T$ with respect to $X_2Y_2Z_2$ -system. Essentially what we have done here is just created a new system with the same orientation as that of $X_1Y_1Z_1$ -system and translated this to a new position C_p^2 with respect to the $X_1Y_1Z_1$ -system only in the X_1Y_1 -plane of the $X_1Y_1Z_1$ -system [Fig. 2.6]. Now the line joining P and the center of the lens meets the X_2Y_2 plane at a point whose $X_2Y_2Z_2$ coordinates are $(X_2, Y_2, 0)^T$. From the geometry shown in Fig. 2.5,

$$X_2 = x_2^p \frac{(\lambda - Z_2)}{\lambda} \quad (2.8)$$

$$Y_2 = y_2^p \frac{(\lambda - Z_2)}{\lambda}. \quad (2.9)$$

At this stage we have two images of the same scene (3-D point) with its projections in the two coordinate systems discussed above.

Since the translation of $X_2Y_2Z_2$ -system was only in the X_1Y_1 -plane of the $X_1Y_1Z_1$ -system we have,

$$Z_1 = Z_2$$

$$X_2 = X_1 - d_1$$

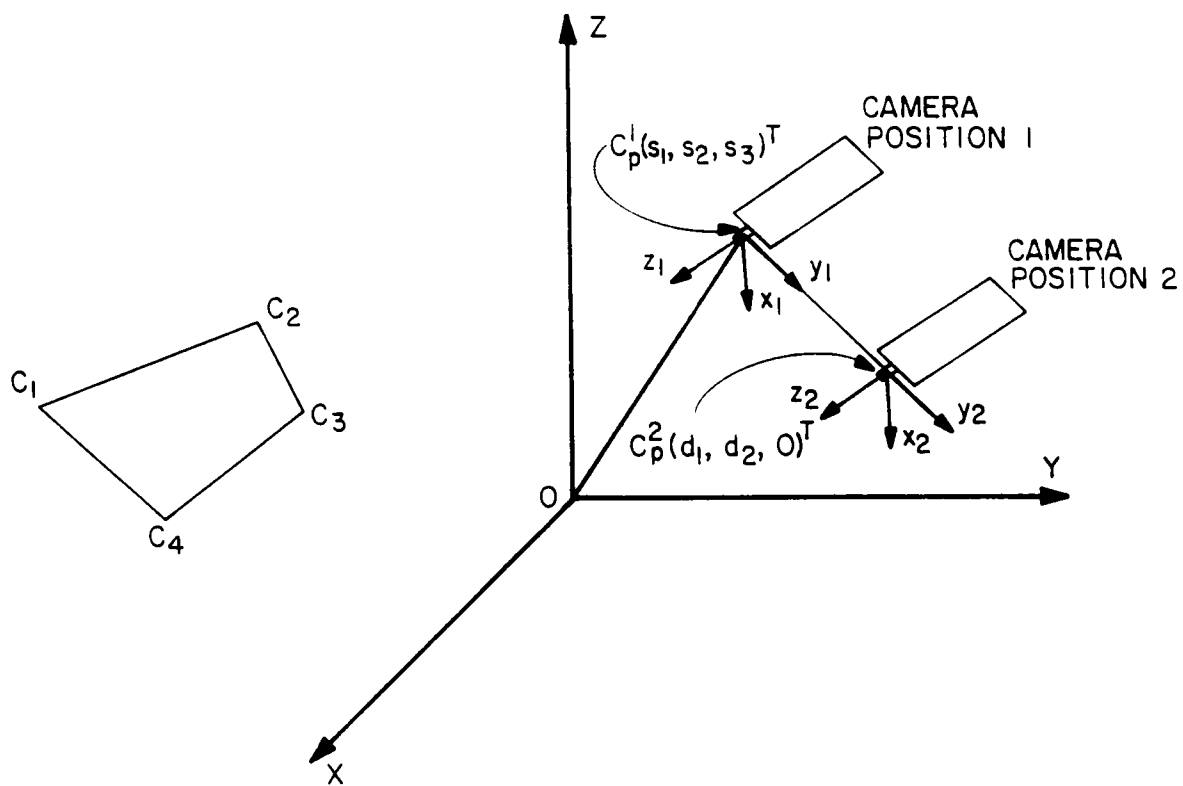


Figure 2.6: Relative camera positions.

$$Y_2 = Y_1 - d_2.$$

Using Eqs. 2.6 and 2.8 we obtain

$$\begin{aligned} X_2 &= x_2^p \frac{(\lambda - Z_2)}{\lambda} = X_1 - d_1 = x_1^p \frac{(\lambda - Z_1)}{\lambda} - d_1 \\ x_2^p \frac{(\lambda - Z_2)}{\lambda} &= x_1^p \frac{(\lambda - Z_1)}{\lambda} - d_1 \\ Z_2 &= \left(\frac{d_1}{x_2^p - x_1^p} + 1 \right) \lambda. \end{aligned}$$

Substituting this value in Eqs. 2.6 and 2.8 we get

$$X_1 = x_1^p \left(\frac{d_1}{x_1^p - x_2^p} \right) \quad (2.10)$$

$$Y_1 = y_1^p \left(\frac{d_1}{x_1^p - x_2^p} \right) \quad (2.11)$$

$$Z_1 = \left(\frac{d_1}{x_2^p - x_1^p} + 1 \right) \lambda \quad (2.12)$$

or substituting this value of Z_2 in Eqs. 2.7 and 2.9 we get

$$X_1 = x_1^p \left(\frac{d_1}{y_1^p - y_2^p} \right) \quad (2.13)$$

$$Y_1 = y_1^p \left(\frac{d_1}{y_1^p - y_2^p} \right) \quad (2.14)$$

$$Z_1 = \left(\frac{d_1}{y_2^p - y_1^p} + 1 \right) \lambda. \quad (2.15)$$

Thus, the point P in the $X_1Y_1Z_1$ -system is obtained.

Consider R as the rotation matrix that transforms a given point from one system to another system. The coordinates of the point $P(X_1, Y_1, Z_1)$ in RCS is

then simply given by the following Equation.

$$R \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ 1 \end{bmatrix} + \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2.16)$$

The rotation matrix transforms the (X_1, Y_1, Z_1) point into the (X, Y, Z) system. The coordinates of C_p^1 within the RCS frame of reference is (s_1, s_2, s_3) . A rotation about the Z -axis by an angle α yields the rotation matrix R_α as

$$R_\alpha = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

rotation about the rotated Y -axis by an angle β yields the rotation matrix R_β as

$$R_\beta = \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

and finally the rotation about the rotated Z -axis by an angle θ yields the rotation

matrix R_θ as

$$R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The resultant rotation matrix is

$$R = R_\alpha R_\beta R_\theta$$

$$R = \begin{bmatrix} \cos\alpha \cos\beta \cos\theta - \sin\alpha \sin\theta & -\cos\alpha \cos\beta \sin\theta - \sin\alpha \cos\theta & \cos\alpha \sin\beta & 0 \\ \sin\alpha \cos\beta \cos\theta + \cos\alpha \sin\theta & -\sin\alpha \cos\beta \sin\theta + \cos\alpha \cos\theta & \sin\alpha \sin\beta & 0 \\ -\sin\beta \cos\theta & \sin\beta \sin\theta & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Rewriting Eq. 2.16 as

$$\mathbf{X} = \mathbf{T}\mathbf{X}_1 \quad (2.17)$$

where,

$$\mathbf{T} = \begin{bmatrix} \cos\alpha \cos\beta \cos\theta - \sin\alpha \sin\theta & -\cos\alpha \cos\beta \sin\theta - \sin\alpha \cos\theta & \cos\alpha \sin\beta & s_1 \\ \sin\alpha \cos\beta \cos\theta + \cos\alpha \sin\theta & -\sin\alpha \cos\beta \sin\theta + \cos\alpha \cos\theta & \sin\alpha \sin\beta & s_2 \\ -\sin\beta \cos\theta & \sin\beta \sin\theta & \cos\beta & s_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

is the transformation matrix that transforms a point from the $X_1Y_1Z_1$ -system to RCS, $\mathbf{X}_1$ is the 3-D point $(X_1, Y_1, Z_1, 1)^T$ in the $X_1Y_1Z_1$ coordinate system, and $\mathbf{X}$ is $(X, Y, Z, 1)^T$ the 3-D point in RCS.

Thus the coordinates of the 3-D point P are

$$\begin{aligned}
X &= X_1(\cos\alpha \cos\beta \cos\theta - \sin\alpha \sin\theta) + Y_1(-\cos\alpha \cos\beta \sin\theta - \sin\alpha \cos\theta) + \\
&\quad Z_1(\cos\alpha \sin\beta) + s_1 \\
Y &= X_1(\sin\alpha \cos\beta \cos\theta + \cos\alpha \sin\theta) + Y_1(-\sin\alpha \cos\beta \sin\theta + \cos\alpha \cos\theta) + \\
&\quad Z_1(\sin\alpha \sin\beta) + s_2 \\
Z &= X_1(-\sin\beta \cos\theta) + Y_1(\sin\beta \sin\theta) + Z_1(\cos\beta) + s_3
\end{aligned} \tag{2.18}$$

where, X_1 , Y_1 , and Z_1 are defined in Eqs. 2.13, 2.14, and 2.15. Thus the coordinates of the 3-D point in space are obtained and the depth is recovered.

2.3 Error Analysis

Evaluation of the accuracy of stereoscopic vision is essential for planning robot tasks. This section finds the *relative error* of the X , Y , and Z coordinates in the robot frame of reference.

Consider the X coordinate (Eq. 2.18) of the world point to find its relative error

$$X = x_1^p \left(\frac{d_1}{y_1^p - y_2^p} \right) A + y_1^p \left(\frac{d_1}{y_1^p - y_2^p} \right) B + \left(\frac{d_1}{y_2^p - y_1^p} \right) C + \lambda C + s_1$$

where,

$$\begin{aligned}
A &= \cos\alpha \cos\beta \cos\theta - \sin\alpha \sin\theta \\
B &= -\cos\alpha \cos\beta \sin\theta - \sin\alpha \cos\theta \\
C &= \cos\alpha \sin\beta.
\end{aligned}$$

The total error on the X coordinate is given by the following relation

$$\begin{aligned}
 dX = & \left[\frac{\partial X}{\partial x_1^p} \right] dx_1^p + \left[\frac{\partial X}{\partial y_1^p} \right] dy_1^p + \left[\frac{\partial X}{\partial y_2^p} \right] dy_2^p + \\
 & \left[\frac{\partial X}{\partial \alpha} \right] d\alpha + \left[\frac{\partial X}{\partial \beta} \right] d\beta + \left[\frac{\partial X}{\partial \theta} \right] d\theta + \\
 & \left[\frac{\partial X}{\partial s_1} \right] ds_1 + \left[\frac{\partial X}{\partial \lambda} \right] d\lambda.
 \end{aligned} \tag{2.19}$$

The three Euler angles α, β , and θ are related to the mechanical characteristics of the robot (kinetic errors), involving accuracy in the measurement of the displacement of the end effector with respect to the RCS. The T³-726 being a precision robot, the errors $d\alpha, d\beta$, and $d\theta$ are very small hence the terms associated with these errors are neglected. Also assuming that the errors associated with s_1 (also of kinetic type) and λ are very small the terms associated with these errors may be neglected. Equation 2.19 then reduces to the following

$$dX = \left[\frac{\partial X}{\partial x_1^p} \right] dx_1^p + \left[\frac{\partial X}{\partial y_1^p} \right] dy_1^p + \left[\frac{\partial X}{\partial y_2^p} \right] dy_2^p. \tag{2.20}$$

Considering only the partial derivatives from the above expression the following relations are found

$$\frac{\partial X}{\partial x_1^p} = \left(\frac{d_1 A}{y_1^p - y_2^p} \right) \tag{2.21}$$

$$\frac{\partial X}{\partial y_1^p} = \left[\frac{-x_1^p d_1 A}{(y_1^p - y_2^p)^2} - \frac{y_2^p d_1 B}{(y_1^p - y_2^p)^2} + \frac{\lambda d_1 C}{(y_1^p - y_2^p)^2} \right] \tag{2.22}$$

$$\frac{\partial X}{\partial y_2^p} = \left[\frac{x_1^p d_1 A}{(y_1^p - y_2^p)^2} + \frac{y_1^p d_1 B}{(y_1^p - y_2^p)^2} - \frac{\lambda d_1 C}{(y_1^p - y_2^p)^2} \right]. \tag{2.23}$$

Substituting 2.21, 2.22, and 2.23 in Eq. 2.20 the total error on the X coordinate is

$$dX = \left(\frac{d_1 A}{y_1^p - y_2^p} \right) dx_1^p + \left[\frac{-x_1^p d_1 A}{(y_1^p - y_2^p)^2} - \frac{y_2^p d_1 B}{(y_1^p - y_2^p)^2} + \frac{\lambda d_1 C}{(y_1^p - y_2^p)^2} \right] dy_1^p +$$

$$\left[\frac{x_1^p d_1 A}{(y_1^p - y_2^p)^2} + \frac{y_1^p d_1 B}{(y_1^p - y_2^p)^2} - \frac{\lambda d_1 C}{(y_1^p - y_2^p)^2} \right] dy_2^p.$$

Dividing the above expression on both sides by X ; simplifying and grouping the proper terms gives the total relative error as

$$\frac{dX}{X} = \left[\frac{A}{A + \frac{y_1^p}{x_1^p} B - \frac{\lambda C}{x_1^p d_1} \{d_1 - (y_1^p - y_2^p)\} + \left(\frac{y_1^p - y_2^p}{d_1 x_1^p} \right) s_1} \right] \frac{dx_1^p}{x_1^p} +$$

$$\left[\frac{-x_1^p A - y_2^p B + \lambda C}{(y_1^p - y_2^p) \left[B + \frac{x_1^p}{y_1^p} A - \frac{\lambda C}{y_1^p d_1} \{d_1 - (y_1^p - y_2^p)\} + \left(\frac{y_1^p - y_2^p}{d_1 y_1^p} \right) s_1 \right]} \right] \frac{dy_1^p}{y_1^p} +$$

$$\left[\frac{x_1^p A + y_1^p B - \lambda C}{x_1^p A + y_1^p B - \frac{\lambda C}{d_1} \{d_1 - (y_1^p - y_2^p)\} + \left(\frac{y_1^p - y_2^p}{d_1} \right) s_1} \right] \frac{dy_2^p}{(y_1^p - y_2^p)}.$$

The maximum relative error on the X coordinate is given by the absolute value of the above expression.

$$\Delta_X = \left| \frac{A}{A + \frac{y_1^p}{x_1^p} B - \frac{\lambda C}{x_1^p d_1} \{d_1 - (y_1^p - y_2^p)\} + \left(\frac{y_1^p - y_2^p}{d_1 x_1^p} \right) s_1} \right| \left| \frac{dx_1^p}{x_1^p} \right| +$$

$$\left| \frac{-x_1^p A - y_2^p B + \lambda C}{(y_1^p - y_2^p) \left[B + \frac{x_1^p}{y_1^p} A - \frac{\lambda C}{y_1^p d_1} \{d_1 - (y_1^p - y_2^p)\} + \left(\frac{y_1^p - y_2^p}{d_1 y_1^p} \right) s_1 \right]} \right| \left| \frac{dy_1^p}{y_1^p} \right|$$

$$+ \left| \frac{x_1^p A + y_1^p B - \lambda C}{x_1^p A + y_1^p B - \frac{\lambda C}{d_1} \{d_1 - (y_1^p - y_2^p)\} + \left(\frac{y_1^p - y_2^p}{d_1} \right) s_1} \right| \left| \frac{dy_2^p}{(y_1^p - y_2^p)} \right| \quad (2.24)$$

Similarly refering to Eq. 2.18 the maximum relative error on the Y and the Z coordinates is given by the Eqs. 2.25 and 2.26.

$$\begin{aligned}
\Delta_Y = & \left| \frac{A}{A + \frac{y_1^p}{x_1^p} B - \frac{\lambda C}{x_1^p d_1} \{d_1 - (y_1^p - y_2^p)\} + \left(\frac{y_1^p - y_2^p}{d_1 x_1^p}\right) s_2} \right| \left| \frac{dx_1^p}{x_1^p} \right| + \\
& \left| \frac{-x_1^p A - y_2^p B + \lambda C}{(y_1^p - y_2^p) \left[B + \frac{x_1^p}{y_1^p} A - \frac{\lambda C}{y_1^p d_1} \{d_1 - (y_1^p - y_2^p)\} + \left(\frac{y_1^p - y_2^p}{d_1 y_1^p}\right) s_2 \right]} \right| \left| \frac{dy_1^p}{y_1^p} \right| \\
& + \left| \frac{x_1^p A + y_1^p B - \lambda C}{x_1^p A + y_1^p B - \frac{\lambda C}{d_1} \{d_1 - (y_1^p - y_2^p)\} + \left(\frac{y_1^p - y_2^p}{d_1}\right) s_2} \right| \left| \frac{dy_2^p}{(y_1^p - y_2^p)} \right|
\end{aligned} \tag{2.25}$$

where,

$$\begin{aligned}
A &= \sin \alpha \cos \beta \cos \theta + \cos \alpha \sin \theta \\
B &= -\sin \alpha \cos \beta \sin \theta + \cos \alpha \cos \theta \\
C &= \sin \alpha \sin \beta
\end{aligned}$$

$$\begin{aligned}
\Delta_Z = & \left| \frac{A}{A + \frac{y_1^p}{x_1^p} B - \frac{\lambda C}{x_1^p d_1} \{d_1 - (y_1^p - y_2^p)\} + \left(\frac{y_1^p - y_2^p}{d_1 x_1^p}\right) s_3} \right| \left| \frac{dx_1^p}{x_1^p} \right| + \\
& \left| \frac{-x_1^p A - y_2^p B + \lambda C}{(y_1^p - y_2^p) \left[B + \frac{x_1^p}{y_1^p} A - \frac{\lambda C}{y_1^p d_1} \{d_1 - (y_1^p - y_2^p)\} + \left(\frac{y_1^p - y_2^p}{d_1 y_1^p}\right) s_3 \right]} \right| \left| \frac{dy_1^p}{y_1^p} \right| \\
& + \left| \frac{x_1^p A + y_1^p B - \lambda C}{x_1^p A + y_1^p B - \frac{\lambda C}{d_1} \{d_1 - (y_1^p - y_2^p)\} + \left(\frac{y_1^p - y_2^p}{d_1}\right) s_3} \right| \left| \frac{dy_2^p}{(y_1^p - y_2^p)} \right|
\end{aligned} \tag{2.26}$$

where,

$$\begin{aligned}
A &= -\sin \beta \cos \theta \\
B &= \sin \beta \sin \theta \\
C &= \cos \beta.
\end{aligned}$$

Equations 2.24, 2.25, and 2.26 represent the correspondence errors, since dx_1^p , dy_1^p , and dy_2^p denote errors made in the matching between corresponding points in the two images.

2.4 Implementation and Experimental Results

In the previous sections of this chapter, we have described a stereo algorithm to extract the X , Y , and Z coordinates of a point in 3-D space along with the relative error associated with it. In this section, we describe an implementation of this algorithm with a numerical example (real data). The algorithm was implemented in C and FORTRAN languages on a VAX 11/785 computer. The image size used in these experiments is 256×256 .

2.4.1 Implementation of the Algorithm

The *vision* system consists of a Fairchild CCD-3000 camera interfaced to a Perceptics 9200e image processor. The image processor has the capability to digitize, store, and process individual frames from the camera video output. The camera is mounted on the end effector of the robot [Fig. 2.7].

To simplify the image processing requirements for our experiments we use a black quadrangular plane as our object with white as a background. The intent of this experiment is to find the position and orientation of this plane using stereoscopic images. The registration problem in the two images was solved by detecting the corners of the object in the two images.

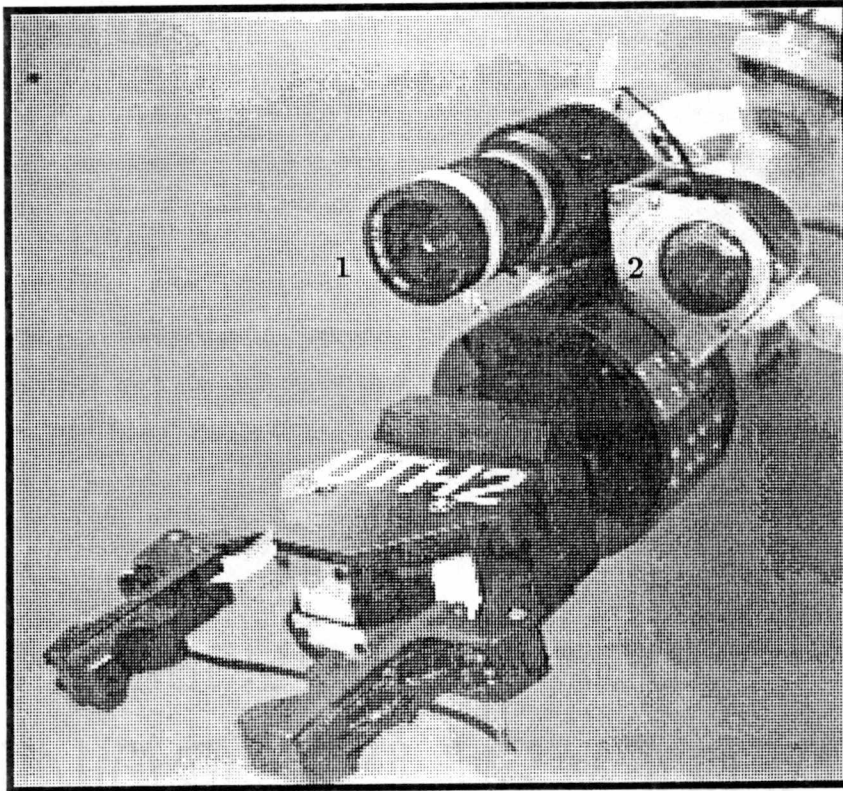


Figure 2.7: Sensors mounted on the end effector of the robot: (1) Vision sensor, (2) Range sensor

The camera was moved to a specific position in the robots environment such that the object was in its field of view. An image (image 1) is taken and preprocessed to find the boundary of the object. Preprocessing was achieved by using the built-in functions on the Perceptics image processor. All the four corners were detected by fitting a straight line through the four edges in the piecewise *least square* sense [16]. The intersection points of these four lines give the pixel coordinates of the four corners. For each possible point a correlation with the raster image was made using a mask. The camera was then moved to a new position with the restrictions discussed in section 2.2.2. The corners in pixel coordinates were acquired from this image (image 2). The image processing steps are shown in Fig. 2.8.

The pixel coordinates were converted into millimeter (*mm*) coordinates by fitting them to a third order polynomial whose coefficients are the camera parameters. The camera parameters were obtained experimentally. The use of the third order polynomial takes into account the camera lens distortion towards the edges of the lens. The polynomial is of the following form with x and y denoting the sensor coordinates of the i, j pixel coordinates in *mm*.

$$\begin{aligned} x_{mm} &= a_1 i^3 + a_2 i^2 j + a_3 i j^2 + a_4 j^3 + a_5 i^2 + a_6 i j + a_7 j^2 + a_8 i + a_9 j + a_{10} \\ y_{mm} &= b_1 i^3 + b_2 i^2 j + b_3 i j^2 + b_4 j^3 + b_5 i^2 + b_6 i j + b_7 j^2 + b_8 i + b_9 j + b_{10} \end{aligned}$$

where, $a_1, \dots, a_{10}$ and $b_1, \dots, b_{10}$ are the camera parameters which are tabulated in Table 2.1. Table 2.2 shows the image coordinates and the *mm* coordinates.

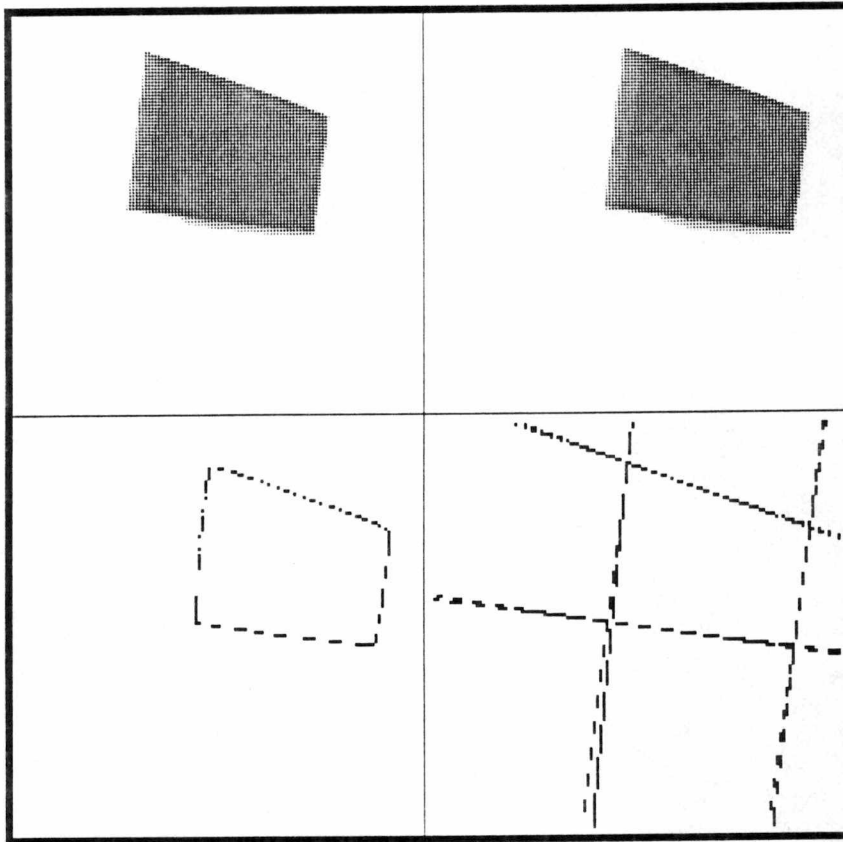


Figure 2.8: Image Processing: Top row shows the left and the right images: bottom left is the boundary of the right image: bottom right shows the corners of the object detected by the intersection of the four lines

Table 2.1: Camera Parameters

i	a_i	b_i
1	$-2.02095188\text{e}-10$	$1.388319590\text{e}-11$
2	$1.28959966\text{e}-11$	$1.83577044\text{e}-10$
3	$-1.86102286\text{e}-10$	$-1.14796247\text{e}-11$
4	$-2.11287900\text{e}-12$	$1.560345460\text{e}-10$
5	$1.42941668\text{e}-07$	$-6.88390779\text{e}-08$
6	$4.00374305\text{e}-08$	$-8.54370484\text{e}-08$
7	$4.787635117\text{e}-08$	$-7.98788729\text{e}-08$
8	$-7.36849463\text{e}-04$	$3.053516434\text{e}-05$
9	$-1.01921569\text{e}-05$	$6.98404751\text{e}-04$
10	0.18321856	-0.1773480178

Table 2.2: Object corner coordinates in pixels and mm

Image 1		Image 2	
coordinates in pixels (x_1^p, y_1^p)	coordinates in mm	coordinates in pixels (x_2^p, y_2^p)	coordinates in mm
(24,83)	(3.81,-1.52)	(24,121)	(3.81,-0.25)
(66,194)	(2.29,2.31)	(66,233)	(2.29,3.81)
(138,184)	(-0.25,1.78)	(139,222)	(-0.25,3.30)
(123,75)	(1.78,-1.78)	(124,112)	(0.15,-0.51)

The formulae derived in Eqs. 2.13, 2.14, 2.15 were implemented to find the 3-D coordinates of these corners in the $X_1Y_1Z_1$ system. Applying Eq. 2.17 these points were transformed into RCS frame of reference. From these values the center points of the edges computed in both the $X_1Y_1Z_1$ and the RCS system are tabulated in Table 2.3.

The next step in the experiments was to find the relative error in the measurements. There are errors induced during digitization, thresholding the gray level image and converting into a binary image, and detection of corners in both the images. Also, since the $X_2Y_2Z_2$ -system (position of the camera for Image 2) was moved only along the Y_1 axis, with respect to the $X_1Y_1Z_1$ -system (position of the camera for Image 1) the x coordinate in the second image should be the same as the x coordinate in the first image. But as seen from Table 2.2 there is an error of 1 or 2 pixels. Assuming the total error to be 7 pixels in both the x, y direction, their equivalent values in mm was found and are tabulated in Table 2.4. Further, Eqs. 2.24, 2.25, and 2.26 were implemented to find the relative error in percentage and tabulated in Table 2.5.

Table 2.3: The 3-D coordinates of the object

$X_1 Y_1 Z_1$ system (X_1, Y_1, Z_1)	XYZ system (X, Y, Z)
(-8.4,3.6,30.2)	(33.2,-22.7,2.11)
(-4.8,-4.9,28.4)	(24.4,-22.2,-1.13)
(0.8,-4.3,29.4)	(24.9,-22.1,-6.86)
(-0.4,4.3,31.3)	(33.8,-22.3,-5.99)

Table 2.4: The correspondence errors corresponding to the 3-D points

corners	correspondence errors		
	∂x_1^p	∂y_1^p	∂y_2^p
1	0.254	0.244	0.241
2	0.254	0.249	0.254
3	0.254	0.244	0.251
4	0.249	0.241	0.241

Table 2.5: Percent relative errors of the 3-D points in RCS

corners	Δ_X	Δ_Y	Δ_Z
1	12	52	80
2	2	49	23
3	2	50	36
4	12	52	37

CHAPTER 3

Range Sensing

In this chapter we begin with a brief introduction to range sensing in section 3.1. In section 3.2 we develop a model for our range sensor. Section 3.3 shows how the vision data is used to position the range sensor relative to the sensed object. Section 3.4 finds the 3-D position of the edge points from the range data. Section 3.5 discusses the implementation of these techniques to acquire range data. Finally the conclusion is given in section 3.6.

3.1 Introduction

Range sensors measure a distance from an object to the sensor. Most of these devices operate by transmitting an optical beam (usually laser) or ultrasonic pulses and then detecting the reflected signal. Some systems use a continuous signal and determine the distance based on the phase relation between the transmitted signal and the reflected signal. Time-of-flight systems (ultrasonic range sensing) transmit a pulse and calculate the distance using the delay time before the reflected signal is received.

Ultrasonic sensing has been viewed with considerable interest because it is inexpensive, gives range information, and can potentially give three dimensional

surface information [30].

3.2 Development of the Sensor Model

The polaroid ultrasonic ranging sensor determines the range by means of an ultrasonic echo. This is accomplished with a capacitive transducer consisting of a very thin metalized diaphragm supported over a specially machined back-plate. A short burst of ultrasonic energy is generated electronically, amplified, and transmitted by a transducer. This signal travels through air, is reflected from the target object, and returns to the transducer. This signal is received, amplified, and processed by the system electronics. The time for the round trip is determined and the distance to the object is calculated.

This type of sensing has met with limited success in the past for several reasons. First, since the acoustic impedance of the air is quite low (sound waves can freely travel in air) and since typical acoustic impedances of solid objects are much larger, all solid surfaces appear as acoustic reflectors. Also, since the acoustic wavelengths are generally quite long, most surfaces appear to be acoustic mirrors. Consequently, surfaces that are not orthogonal to the direction of acoustic propagation reflect signal energy away from the source, and the surface is not detectable. Furthermore, from a typical ultrasonic sensor, the problem of acoustic beam expansion of the transmitted signal allows return echoes to be obtained from any angle inside the cone of propagation. Also the

magnitude of the range depends on the velocity of the sound in ambient air, which is affected by air currents and temperature [31]. Under these conditions, smooth flat surfaces can appear to be quite rough to a scanning ultrasonic sensor.

The typical beam pattern of a 50kHz beam at approximately 15 degrees of the ranging unit is as shown in Fig. 3.1. This pattern was verified experimentally. As seen, it is very difficult to model this graph mathematically. This along with the problem of the sensor being orthogonal to the object plane lead us to model our sensor experimentally. The problem of expansion of the acoustic beam was reduced by installing a cone on the sensor diaphragm [Fig. 3.2]. This cone would decrease the beam width and thereby help us in the detection of the object edges with a higher confidence than without the cone. The effects of air and temperature were neglected.

An empirical model of the sensor was created for varied degrees of orientation (θ) of the sensor to the object and the depth (d in inches) from the sensor to the object. For a fixed θ , d varying, the object was scanned to detect its edge point. This process was repeated for different values of θ . The process of building the model was divided into three parts. The first part dealt with finding the percentage error ϵ_1 , with the cone of propagation making an angle α . For different values of d and θ , ϵ_1 was computed in percent as the ratio of the error in the measured depth to the true depth. A graph was plotted for different values of θ and d . This is illustrated in Fig. 3.3(b). In this plot θ_α denotes that this plot is for the error associated with the angle α made by the

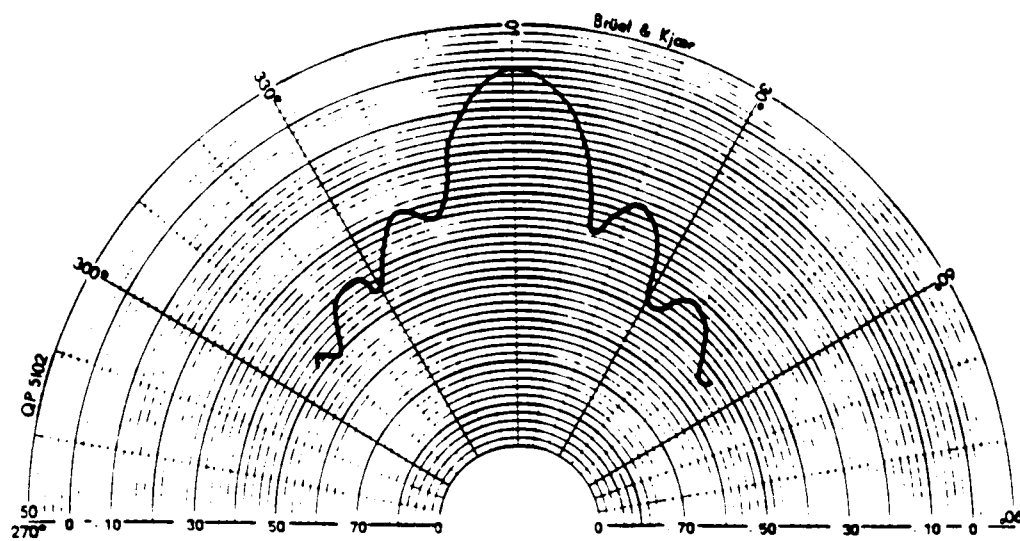


Figure 3.1: Typical Beam Pattern (reprinted from the technical manual supplied by the manufacturer)

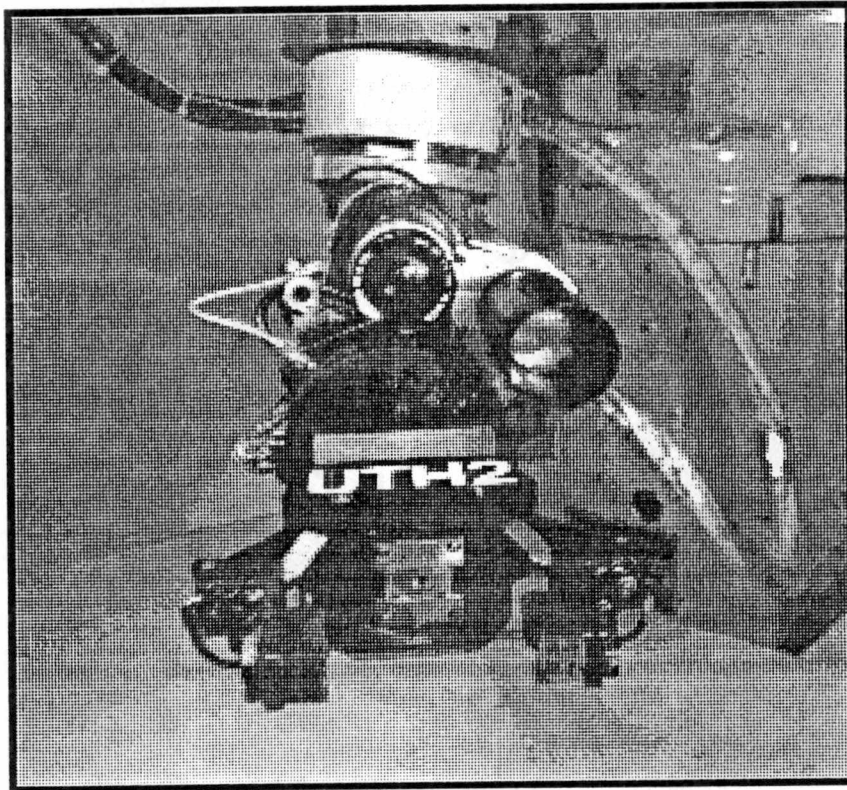


Figure 3.2: Range sensor with the cone

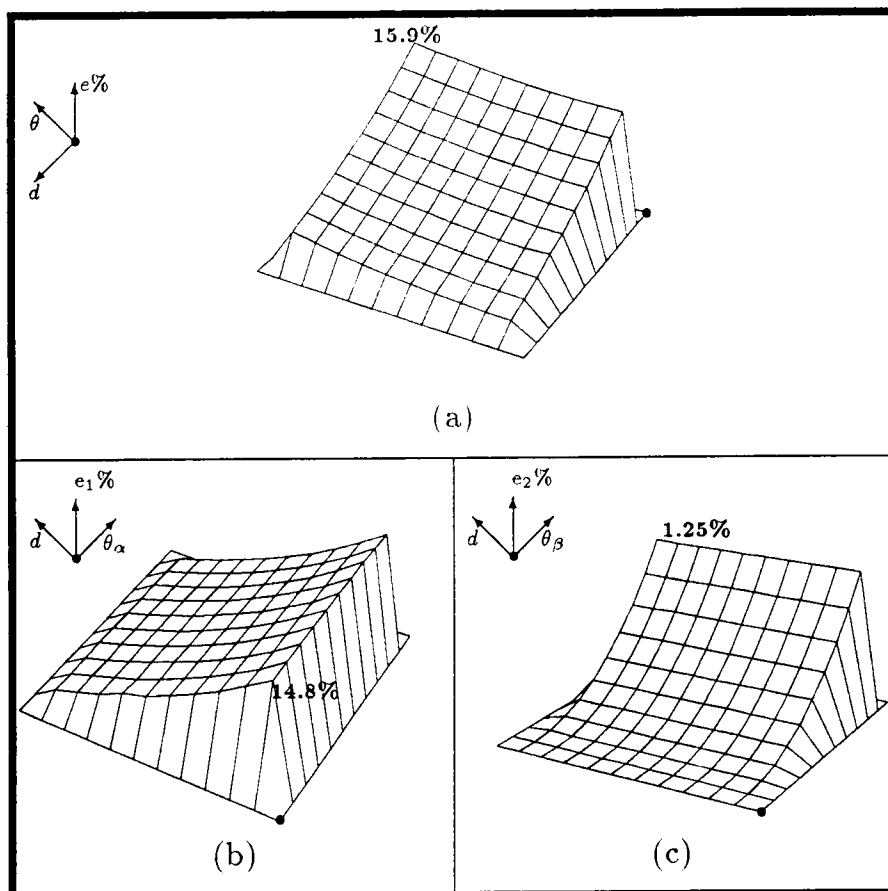


Figure 3.3: (a): The sensor model; (b): percentage error with the cone of propagation making an angle α ; (c): percentage error when the sensor makes an angle θ with respect to the object

cone of propagation. The second part finds the error in estimating the distance from the sensor to the object when the sensor is making an angle θ with the object (neglecting the cone of propagation). This was achieved by varying θ and measuring the distance from the sensor to the object for different depths d . For different measurements the error was computed in percent as the ratio of error to the true distance. This is illustrated in Fig. 3.3(c). In this plot, θ_3 denotes the orientation of the sensor to the object neglecting the cone of propagation. Finally, in the third part, these two errors are combined together to form the range sensor model and is illustrated in Fig. 3.3(a). For clarity the plots in Fig. 3.3 show the viewing coordinates for the respective angles and the numbers in percentage indicate the maximum error in each of the cases. The data points for the range model are tabulated in Table 3.1. Looking at the plots and the data table it is seen that for a fixed θ , the percentage error decreases as the depth increases, but as θ increases the error increases for a fixed depth and for an increasing depth, *i.e.*, the error is a function of d and θ . So it was decided that the measurements would be acceptable if the orientation of the sensor with respect to the object does not exceed 9 degrees and the depth of the sensor to the object should be around 20 inches.

Table 3.1: Range Sensor model data: the elements of the matrix represent $\epsilon\%$.

$d \setminus \theta$	0°	1°	2°	3°	4°	5°	6°	7°	8°	9°
$15''$	14.8	14.8	14.9	14.9	15.0	15.2	15.3	15.6	15.7	15.9
$16''$	13.9	13.9	14.0	14.0	14.2	14.3	14.4	14.6	14.8	15.0
$17''$	13.1	13.1	13.2	13.3	13.3	13.5	13.6	13.8	14.0	14.2
$18''$	12.4	12.4	12.4	12.5	12.6	12.8	12.9	13.1	13.3	13.5
$19''$	11.7	11.8	11.8	11.9	12.0	12.1	12.3	12.4	12.7	12.9
$20''$	11.1	11.2	11.2	11.3	11.4	11.5	11.7	11.9	12.1	12.3
$21''$	10.8	10.7	10.7	10.8	10.9	11.0	11.2	11.3	11.6	11.8
$22''$	10.2	10.2	10.2	10.3	10.4	10.5	10.7	10.9	11.1	11.3
$23''$	9.7	9.7	9.8	9.9	10.0	10.1	10.2	10.4	10.7	10.9
$24''$	9.3	9.3	9.4	9.5	9.6	9.7	9.8	10.0	10.3	10.5

3.3 Intensity Guided Range Sensing

In the previous chapter we found the position of the four corners of the quadrangle using stereo imaging technique. It is very difficult to make range measurements on corners of any object, hence it was decided to make range measurements on the mid-points of the two corners which would detect the edge of the object. This section is divided in two subsections. In subsection 3.3.1 we find the midpoints of these corners and try to recover the corners from these midpoints under some restrictions. Subsection 3.3.2 shows how to orient the range sensor such that it is normal to the object using the vision data.

3.3.1 Recovery of corners

Consider the four corners C_1, C_2, C_3 , and C_4 of the quadrangle as shown in Fig. 3.4. The midpoints M_1, M_2, M_3 , and M_4 of the corners can be easily

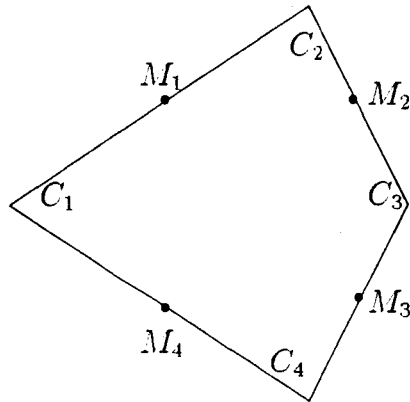


Figure 3.4: Quadrangle with its corners and midpoints

obtained using the mid-point formula. Knowing the mid-points we attempt to recover the corners. Again using the mid-point formula we have the following relationship for the x -coordinate of the midpoints.

$$2M_{1x} = C_{1x} + C_{2x}$$

$$2M_{2x} = C_{2x} + C_{3x}$$

$$2M_{3x} = C_{3x} + C_{4x}$$

$$2M_{4x} = C_{4x} + C_{1x}$$

Writing the above relation in matrix form yields

$$\underbrace{\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}}_A \begin{bmatrix} C_{1x} \\ C_{2x} \\ C_{3x} \\ C_{4x} \end{bmatrix} = 2 \begin{bmatrix} M_{1x} \\ M_{2x} \\ M_{3x} \\ M_{4x} \end{bmatrix}. \quad (3.1)$$

The last row of A in Eq. 3.1 is a linear combination of the first three rows of A which means that A is a *singular* matrix. This suggests there is no unique solution to the problem. Knowing the mid-points is not sufficient to recover the corners of the quadrangle. The midpoints of a convex¹ quadrangle always form a parallelogram.

By fixing one parameter in Eq. 3.1, and knowing one of the corner points say C_1 , the solution to the problem is given by the following relation which is

¹This is a necessary but not sufficient condition.

written in a matrix form.

$$\begin{bmatrix} C_{1x} \\ C_{2x} \\ C_{3x} \\ C_{4x} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 2 & 0 & 0 \\ 1 & -2 & 2 & 0 \\ -1 & 2 & -2 & 2 \end{bmatrix} \begin{bmatrix} C_{1x} \\ M_{1x} \\ M_{2x} \\ M_{3x} \end{bmatrix}. \quad (3.2)$$

Equation 3.2 gives the x -coordinate of the four corners. The y and z -coordinates can be found similarly.

3.3.2 Orientation of the Range Sensor

Given a point in space its orientation can be found by determining the orientation of the vector normal to the point in space with respect to some fixed coordinate system.

Consider the corner C_3 [Fig. 3.4] as some point in 3-D space. The normal $\hat{n}$ to this point is the cross product of the two vectors C_1C_3 and C_2C_3 , *i.e.*,

$$\hat{n} = (C_1\vec{C}_3 \times C_2\vec{C}_3)$$

Let α and β represent the rotation angles about the Z and the Y -axes, respectively. Referring to Fig. 3.5 the orientation of this vector is given by the following relations

$$\alpha = \arctan(y/x)$$

$$\beta = \arctan(\sqrt{x^2 + y^2}/z)$$

where, x, y, z are the coefficients of the unit normal.

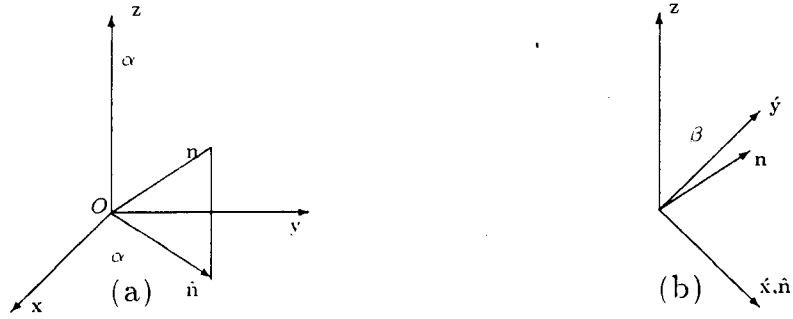


Figure 3.5: Orientation Angles: (a) rotation about the z -axis; (b) rotation about the y -axis

Thus the orientation of C_3 is obtained in 3-D space with respect to a fixed system. Since C_3 lies on the plane $C_1C_2C_3C_4$ the orientation of C_3 is the orientation of the plane. Aligning the range sensor to this orientation makes the sensor normal to the sensed plane.

3.4 3-D Position from Range Data

The profile of range data is as shown by curve A in Fig. 3.6. Curve B in Fig. 3.6 represents an ideal curve for the range sensor if an edge of the object was detected. Assume t (shown in Fig. 3.7) as the distance from where the object was scanned to the point where the object edge was located. The objective here is to find this distance t from our sensory data.

It is assumed that after smoothing curve A in Fig. 3.6 would resemble the integral of the normal distribution curve. This smooth curve is defined by the

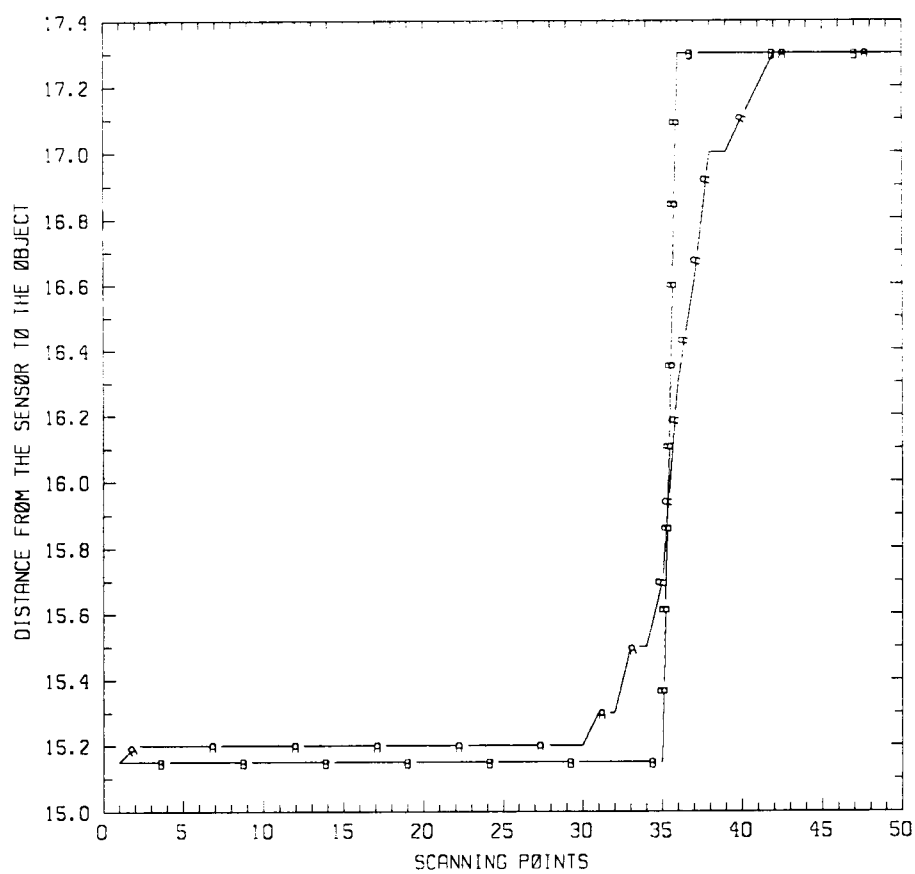
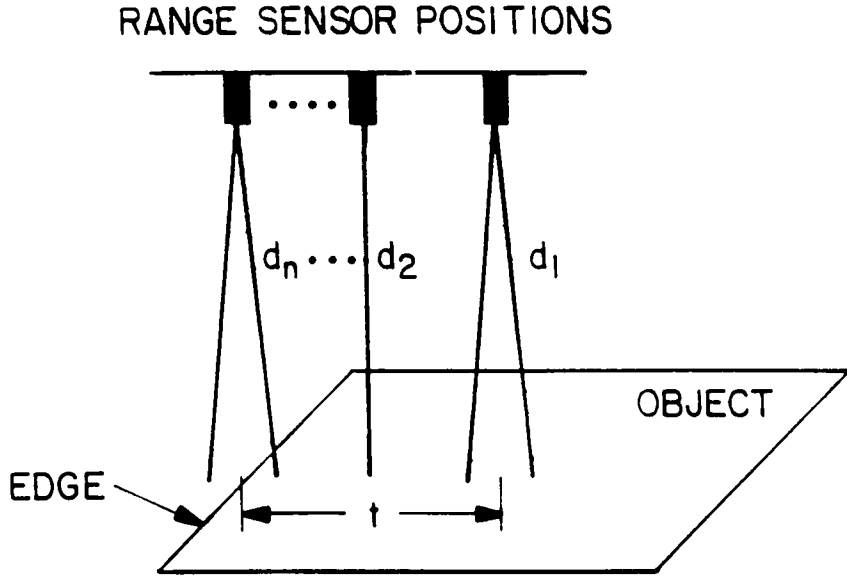


Figure 3.6: Profile of the Range data: curve A represents the actual data set. curve B represents the ideal data set



error function given by Eq. 3.3.

$$\text{erf}(u) = \int_{-\infty}^u \frac{1}{\sqrt{2\pi}\sigma_1} \exp^{-1/2(x-m_1)^2/\sigma_1^2} dx \quad (3.3)$$

The smoothing operation is performed by convolving the sensory data (curve A) with a normal distribution having a zero mean and known variance. Differentiating this convolved curve will yield a normal distribution curve with m_1 as its mean and a new variance σ_1 . This is true since the convolution of two normal distributions $N_1(m_1, \sigma_1)$ and $N_2(m_2, \sigma_2)$ respectively yields a normal distribution $N(M, \sigma)$, where, $M = (m_1 + m_2)$ and $\sigma = (\sigma_1^2 + \sigma_2^2)$ [32].

Thus, for zero mean m_2 , t is the mean m_1 with variance $(\sigma_1^2 + \sigma_2^2)$, where, m_1 is the mean and σ_1 is the variance of the data fitted to the normal curve.

$$\begin{aligned}
x &= x_1 + a_1 t \\
y &= y_1 + a_2 t \\
z &= z_1 + a_3 t
\end{aligned}
\tag{3.4}$$

Knowing t and a line parallel to the line on the plane passing through a known point on the plane, the 3-D coordinates of the edge point can be easily found using the parametric equations of a line in space as shown in Eq. 3.4 (with reference to Fig. 3.8).

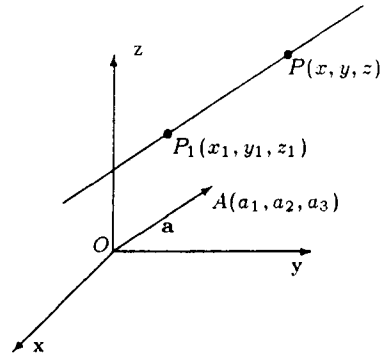


Figure 3.8: Lines in Space

3.5 Implementation and Experimental Results

Previous sections of this chapter described the sensor model and how the range sensor can be guided using the stereo data to acquire the range data. This section implements this theory and acquires the 3-D positions of the four

mid-points from the range data.

Using the stereo data the range sensor was moved to a known position (approximately the center of the plane). The plane was scanned for each edge point. The range sensor traversed parallel and normal to the plane. The data for one of the edge points is as plotted in Fig. 3.9(a). This data was convoluted with a Gaussian distribution function with a variance of 3. This plot is shown in Fig. 3.9(b) and the result of this convolution is shown in Fig. 3.9(c). This data plot now resembles that of an error function (Eq. 3.3). This data is numerically differentiated to get an approximate normal distribution of the data and is as shown in Fig. 3.9(d). This data was fitted to a normal distribution and its mean and variance calculated. This *mean* is t which is the parameter in the parametric representation of a line in space as discussed in section 3.4. The parameter t with its uncertainty and variance for all the four edge points is shown in Table 3.2.

Table 3.2: The data set parameters

Edge No.	t	variance	error in %
1	3.50	5.99	0.15
2	4.75	18.77	0.28
3	3.63	6.54	0.15
4	3.63	5.38	0.14

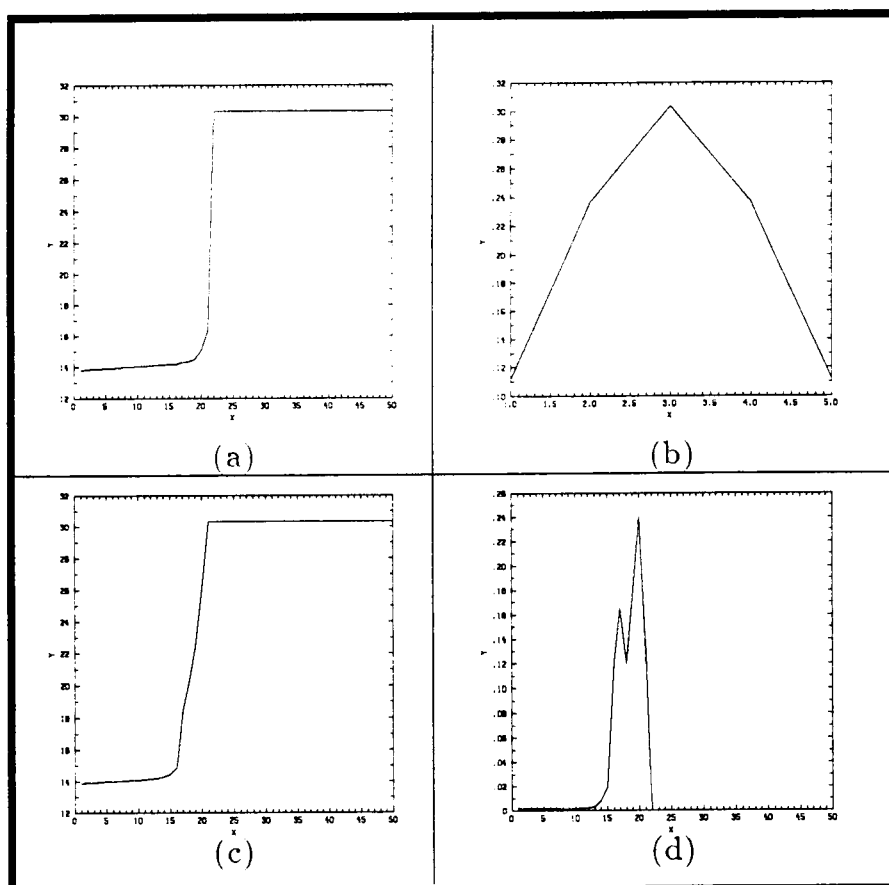


Figure 3.9: (a): Sensor data; (b): Discrete normal distribution with variance 3; (c): Convolution of (a) & (b); (d): Numerical differentiation of (c)

The 3-D coordinates of all the edge points were computed using Eq. 3.4. Their values and the associated variances are tabulated in Table 3.3.

Table 3.3: 3-D points from range data

Edge No.	Coordinates	Variance
1	(28.9, -22.9, 0.45)	4.3
2	(24.1, -21.9, -4.0)	20.6
3	(31.4, -22.2, -5.3)	21.3
4	(34.2, -22.5, -1.9)	5.1

3.6 Conclusion

The 3-D position and orientation for a quadrangular plane as the object in the robots environment was found using stereo and intensity guided range sensing. The errors were computed and tabulated.

CHAPTER 4

Sensor Fusion

This chapter addresses the issue of fusion of sensory data from stereo and ultrasonic range to systematically deal with uncertainty inherent in both sensory data. Section 4.1 briefly formulates the *joint density function* of random variables. Section 4.2 models the measurements as density functions. Section 4.3 formulates the fusion process. Section 4.4 gives the numerical solution of the fusion process. Section 4.5 presents implementation and experimental results.

4.1 Joint Density Functions

The *central limit theorem* [32] states that if the random variables $\mathbf{x}_i$ are independent, then the density of their sum, $\mathbf{x} = \mathbf{x}_1 + \cdots + \mathbf{x}_n$ tends to be a normal curve as $n \rightarrow \infty$. Assuming that the measurements made have a normal density function, their joint probability density functions (p.d.f) are computed. The following subsections give a brief description of computing the density functions of sum, difference, product, and ratio of two random variables.

4.1.1 p.d.f: Function of a Random Variable

Consider $y = f(x)$ where x is a random variable (RV). Then the density function of y is given by

$$p(y) = \frac{p(x)|_{(x=f(y))}}{|f'(x)|} \quad (4.1)$$

where, $p(x)$ is the density function of x and $f'(x)$ is the derivative of $f(x)$.

Case 1: Consider

$$y = ax$$

where, a is a constant then the density function of y using Eq. 4.1 is given by

$$p(y) = \frac{1}{|a|} p\left(\frac{y}{a}\right). \quad (4.2)$$

Case 2: Consider

$$y = ax + b$$

where, a and b are constants then the density function of y is

$$p(y) = \frac{1}{|a|} p\left(\frac{y-b}{a}\right). \quad (4.3)$$

Case 3: Consider

$$y = \frac{1}{x}.$$

The equation $y = 1/x$ has a single solution $x = 1/y$, thus the density function of y is

$$p(y) = \frac{1}{y^2} p\left(\frac{1}{y}\right). \quad (4.4)$$

4.1.2 p.d.f: Difference of Two Random Variables

Consider two independent RVs x and y with known density functions $p(x)$ and $p(y)$ respectively. Let

$$z = x - y.$$

If the two RVs are independent, the density of their sum z equals the convolution of their densities. Thus, the density function of z is given by

$$p(z) = \int_{-\infty}^{\infty} p(z + y)p(y)dy. \quad (4.5)$$

4.1.3 p.d.f: Ratio of Two Random Variables

Consider two independent RVs x and y with known density functions $p_x(x)$ and $p_y(y)$ respectively and let z be the ratio of these two random variables

$$z = \frac{x}{y},$$

then the p.d.f of x and y is given by

$$p_z(z) = \int_{-\infty}^{\infty} |y|p(yz, y)dy. \quad (4.6)$$

where $p(., .)$ is the joint probability density of x and y .

4.1.4 p.d.f: Product of Two Random Variables

Consider two independent RVs x and y , with known density functions $p_x(x)$ and $p_y(y)$ respectively and let z be the product of these two random variables:

$$z = xy.$$

The p.d.f of z is given by

$$p_z(z) = \int_{-\infty}^{\infty} \frac{1}{|y|} p(z/y, y) dy. \quad (4.7)$$

where $p(.,.)$ is the joint probability density of x and y .

4.2 Modelling of Density Functions

Assume that the measurements y_1^p and y_2^p to have normal density functions $p(y_1^p) \simeq N(\mu_1, \sigma)$ and $p(y_2^p) \simeq N(\mu_2, \sigma)$ respectively with the same variance.

The p.d.f of the difference of two RVs is given by

$$p(z) = \frac{1}{2\pi\sigma^2} \int_{-\infty}^{\infty} \exp\left\{-(z+y-\mu_1)^2/2\sigma^2\right\} \exp\left\{-(y-\mu_2)^2/2\sigma^2\right\} dy.$$

Solving the above integral yields

$$p(z) = \frac{1}{\sqrt{2\pi}(\sqrt{2}\sigma)} \exp\left\{-(z-(\mu_1-\mu_2))^2/2(\sqrt{2}\sigma)^2\right\}. \quad (4.8)$$

Equation 4.8 suggests that the joint density function of the difference of two random variables having normal density functions with different means but the same variance is a normal density function. Its mean is the difference of the two means and has the same variance.

Assume x_1^p to have a normal density function $N(\mu_3, \sigma_1)$, and $y = y_1^p - y_2^p$ to have the density function as defined in Eq. 4.8, the p.d.f of x and y using Eq. 4.6 is

$$p(z) = \frac{1}{\pi 2\sqrt{2}\sigma_1\sigma} \int_{-\infty}^{\infty} |y| \exp\left\{-\frac{1}{2\sigma_1^2}(zy - \mu_3)^2\right\} \exp\left\{-\frac{1}{4\sigma^2}(y - (\mu_1 - \mu_2))^2\right\} dy.$$

Solving the above integral yields

$$p(z) = \frac{A}{\alpha} \left[1 + \frac{k\sqrt{\pi} \exp(k^2/4)}{2} \operatorname{erf}(k/2) - 1 \right]. \quad (4.9)$$

with,

$$A = \frac{1}{2\sqrt{2}\pi\sigma^2} \exp(-(2\mu_3^2 + (\mu_1 - \mu_2)^2)/4\sigma^2)$$

$$\alpha = \frac{2z^2 + 1}{4\sigma^2}$$

$$k = \frac{-(4z\mu_3 + 2(\mu_1 - \mu_2))}{2\sigma(\sqrt{2z^2 + 1})}$$

Equation 4.9 gives the p.d.f of the ratio of two independent RVs with $p(x) \simeq N(\mu_3, \sigma_1)$ and $p(y)$ as defined in Eq. 4.8.

Using the above developed relationships, the density functions of X_1 , Y_1 , and Z_1 (Eqs. 2.13,2.14,2.15) which are the joint probability density functions of x_1^p, y_1^p and z_1^p , are given by the following Eqs.

$$\begin{aligned} p^s(X_1) &= \frac{A_1}{\alpha_1 \gamma_1} \left[1 + \frac{k_1 \sqrt{\pi} \exp(k_1^2/4)}{2} (\operatorname{erf}(k_1/2) - 1) \right] \\ p^s(Y_1) &= \frac{A_2}{\alpha_2 \gamma_1} \left[1 + \frac{k_2 \sqrt{\pi} \exp(k_2^2/4)}{2} (\operatorname{erf}(k_2/2) - 1) \right] \\ p^s(Z_1) &= \frac{|\lambda \gamma_1|}{2\sqrt{\pi} \sigma Z_1^2} \exp(-(\gamma_1 \lambda / (Z_1 - \lambda) - (\mu_2 - \mu_1))/2\sigma)^2 \end{aligned} \quad (4.10)$$

with

$$A_1 = \exp(-(2\mu_3^2 + (\mu_1 - \mu_2)^2)/4\sigma^2)/2\sqrt{2}\pi\sigma^2$$

$$A_2 = \exp(-(2\mu_1^2 + (\mu_1 - \mu_2)^2)/4\sigma^2)/2\sqrt{2}\pi\sigma^2$$

$$\alpha_1 = (2X_1^2 + 1)/4\sigma^2$$

$$\alpha_2 = (2Y_1^2 + 1)/4\sigma^2$$

$$k_1 = (-4X_1\mu_3 - 2(\mu_1 - \mu_2))/4\sigma^2\sqrt{\alpha_1}$$

$$k_2 = (-4Y_1\mu_1 - 2(\mu_1 - \mu_2))/4\sigma^2\sqrt{\alpha_2}$$

Equation 4.10 represent the density functions of the 3D points obtained from stereo. The shape of these density functions is illustrated in Fig. 4.1.

The values of σ , σ_1 , μ_1 , μ_2 , μ_3 , and $\lambda_1 = 3$ for all four points of the plane are tabulated in Tables 2.4 and 2.2. respectively. The normal density table for all four points acquired from stereo is as shown in Table 4.1.

Table 4.1: The normal density table of points obtained from stereo (dimensions are in millimeters)

Corner #	$x_1^p \simeq N(\mu_3, \sigma_1)$	$y_1^p \simeq N(\mu_1, \sigma)$	$y_2^p \simeq N(\mu_2, \sigma)$
1	(3.81,0.254)	(-3.05,0.244)	(-1.78,0.244)
2	(3.05,0.254)	(1.78,0.244)	(3.30,0.244)
3	(-0.76,0.254)	(1.52,0.244)	(2.79,0.244)
4	(-2.03,0.254)	(-3.56,0.244)	(-2.29,0.244)

The joint density function for the range data can be formulated as follows:

$$p^r(X_1) = \int_{-\infty}^{\infty} p^s(X_1 - y)p(y)dy$$

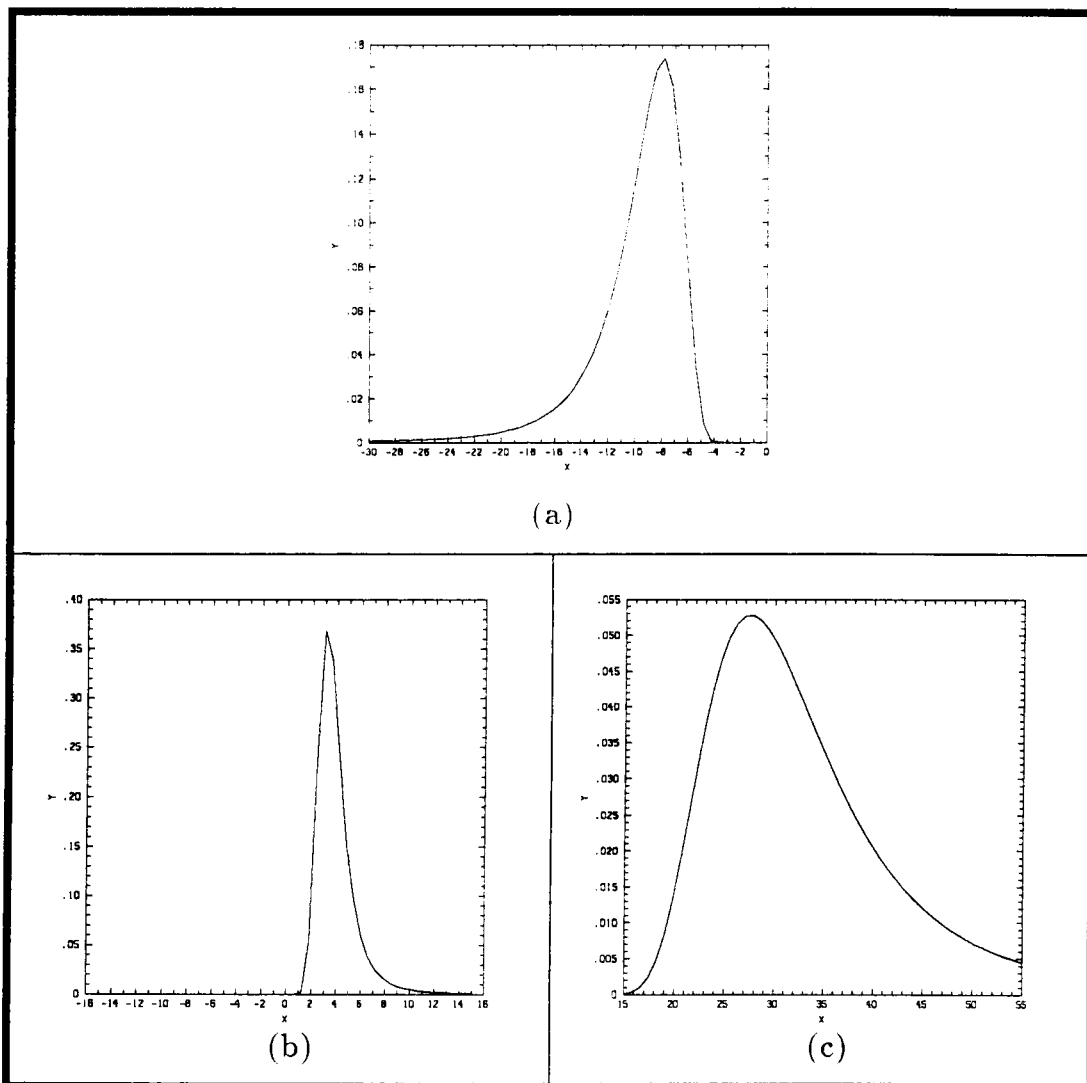


Figure 4.1: Density Functions for Stereo Data: (a): p.d.f $p_{X_1}^s$, (b): p.d.f $p_{Y_1}^s$, (c): p.d.f $p_{Z_1}^s$

$$p^r(Y_1) = \int_{-\infty}^{\infty} p^s(Y_1 - y)p(y)dy \quad (4.11)$$

$$p^r(Z_1) = \int_{-\infty}^{\infty} p^s(Z_1 - y)p(y)dy$$

where,

$$p^s(X_1 - y) = A_1 \left[1 + \frac{k_1 \sqrt{\pi} \exp(k_1^2/4)}{2} (\operatorname{erf}(k_1/2) - 1) \right] / \alpha_1 \gamma_1$$

$$A_1 = \exp(-(2\mu_3^2 + (\mu_1 - \mu_2)^2)/4\sigma^2) / 2\sqrt{2}\pi\sigma^2$$

$$\alpha_1 = (2(X_1 - y)^2 + 1)/4\sigma^2$$

$$k_1 = (-4(X_1 - y)\mu_3 - 2(\mu_1 - \mu_2))/4\sigma^2\sqrt{\alpha_1}$$

$$p(y) = \exp(-(y - \mu_r)^2/2\sigma_r^2)/\sqrt{2\pi}$$

$$p^s(Y_1 - y) = A_1 \left[1 + \frac{k_1 \sqrt{\pi} \exp(k_1^2/4)}{2} (\operatorname{erf}(k_1/2) - 1) \right] / \alpha_1 \gamma_1$$

$$A_1 = \exp(-(2\mu_3^2 + (\mu_1 - \mu_2)^2)/4\sigma^2) / 2\sqrt{2}\pi\sigma^2$$

$$\alpha_1 = (2(Y_1 - y)^2 + 1)/4\sigma^2$$

$$k_1 = (-4(Y_1 - y)\mu_3 - 2(\mu_1 - \mu_2))/4\sigma^2\sqrt{\alpha_1}$$

$$p(y) = \exp(-(y - \mu_r)^2/2\sigma_r^2)/\sqrt{2\pi}$$

$$p^s(Z_1 - y) = \frac{|\lambda\gamma_1|}{2\sqrt{\pi}\sigma(Z_1 - y)^2} \exp(-(\gamma_1\lambda/((Z_1 - y) - \lambda) - (\mu_2 - \mu_1))/2\sigma)^2$$

It can be seen that the analytical solution for these integrals (Eqs. 4.11) would be very complex. Thus a numerical solution for some limits is implemented.

The shape of these density functions is illustrated in Fig. 4.2.

4.3 Fusion of the Density Functions

This section addresses the fusion of two probability density functions derived from the stereo and range data. The density functions are normalized to one as the area under the density curves has to be one. Let $p^s(x)$ and $p^r(x)$ be the two input density functions and $p(x)$ denote the output density resulting from the fusion. This output can be expressed as an unknown functional $\mathcal{F}$ of the two inputs:

$$p(x) = \mathcal{F}(p^s(x), p^r(x)).$$

The focus in the following mathematical formulation is the determination of the fusion functional $\mathcal{F}$ [33]. Assuming that in some neighborhood around $(p^s(x_0), p^r(x_0))$, $\mathcal{F}$ is infinitely differentiable, $\mathcal{F}$ can be expanded in a converging Taylor series:

$$p(x) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} a_{ij} (p^s)^i(x) (p^r)^j(x),$$

where $(p^s)^i(x)$ is the i th power of $p^s(x)$, $(p^r)^j(x)$ is the j th power of $p^r(x)$, and a_{ij} are the unknown coefficients. Each $a_{ij}(x)$ can be expressed as an infinite sum containing partial derivatives of the functional $\mathcal{F}$ of the order $(i + j)$ and higher.

By expanding the functional $\mathcal{F}$ in Taylor series, the problem of determining $\mathcal{F}$ becomes the problem of determining the unknown coefficients of its Taylor series. As the order of truncation increases the complexity in finding the unknown

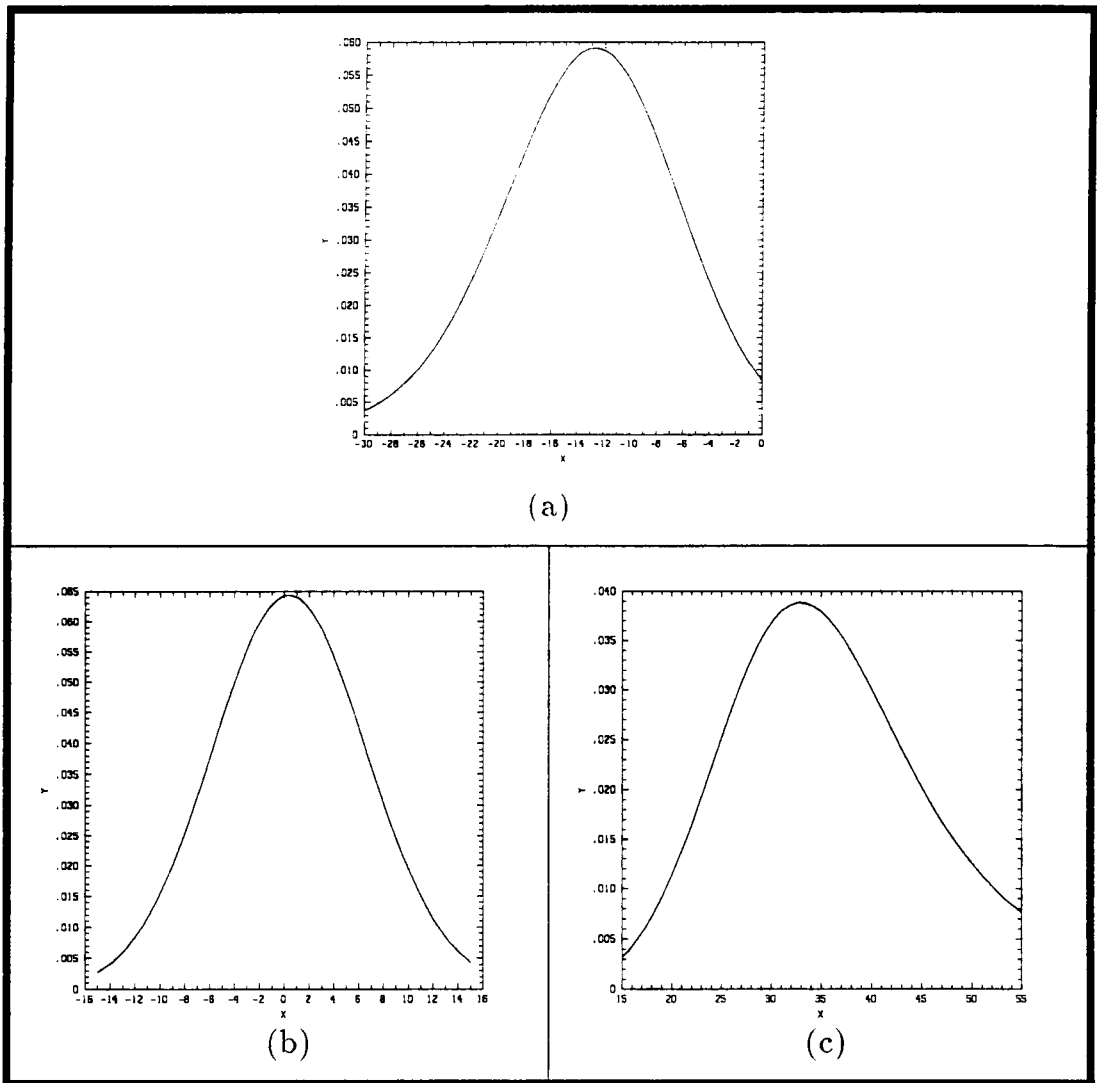


Figure 4.2: Density Functions for Range Data: (a): p.d.f $p_{X_1}^r$, (b): p.d.f $p_{Y_1}^r$, (c): p.d.f $p_{Z_1}^r$

coefficients increases [29] .

4.3.1 First Order Approximation of the Fusion Functional

In the following sections, the dependency upon the independent variable x will be assumed and not explicitly denoted. The first order approximation of $\mathcal{F}$ yields:

$$p = a_{10}p^s + a_{01}p^r, \quad (4.12)$$

which is a linear combination of the inputs.

Equation 4.12 can be normalized so that $0 \leq p \leq 1$. As p^s and p^r are already normalized, this process will affect only the coefficients a_{10} and a_{01} .

$$\begin{aligned} p_{norm} &= (a_{10}p^s + a_{01}p^r)/(a_{10} + a_{01}) \\ &= \alpha p^s + \beta p^r \end{aligned}$$

with $\alpha = a_{10}/(a_{10} + a_{01})$ and $\beta = a_{01}/(a_{10} + a_{01})$. As a result of this scaling α and β are related by the relation: $\alpha + \beta = 1$.

For the remainder of this mathematical formulation, the normalized expression will be of interest and the reference to the normalization can be dropped:

$$p = \alpha p^s + \beta p^r = \alpha p^s + (1 - \alpha)p^r.$$

The relation $\alpha + \beta = 1$ reduces the problem of determining two unknown coefficients to that of determining one unknown coefficient, namely α . The following two constraints lead to the determination of α .

Fusion Constraints

In order to determine the unknown coefficient α two meaningful constraints can be expressed (α will be left as a function of x to allow it to vary from one interval to another):

A) Sensor Corroboration

The fusion of information provided by two disparate sensors should increase the knowledge about a given feature. Hence, a constraint must exist which allows $\mathcal{F}$ to be significant. The highest value $\mathcal{F}$ can reach is one since it has been normalized. Thus the problem of maximizing $\mathcal{F}$ can be expressed as the minimization of the expression $(1 - p)$. This constraint can be formulated in the least mean square (LMS) sense as follows:

$$\int_C (1 - p)^2 dx \quad \text{Minimum.}$$

The integration is carried out over the entire space of x which will be denoted by C .

Let f denote the function $(1 - p)$

$$f = 1 - p = 1 - \alpha p^s - \beta p^r.$$

Using the normalizing constraint yields

$$f = (1 - p^r) - \alpha(p^s - p^r). \quad (4.13)$$

B) Sensor Validation

Any information about the presence or absence of a given feature provided by sensor 1 must be validated by sensor 2 either by confirming or denying this information. As the two sensors record two fairly similar data of the same scene, some similarity is expected between the two density functions p^s and p^r . Both p^s and p^r are expected to vary at the same location. This similarity in the variation of features can be mathematically expressed as a minimization problem on the derivative of their differences. In the LMS sense this can be written as:

$$\begin{aligned} & \int_C \left[\frac{d}{dx} (\alpha p^s - \beta p^r) \right]^2 dx && \text{Minimum.} \\ & = \int_C g^2 dx \end{aligned}$$

with

$$\begin{aligned} g &= (\alpha p^s - \beta p^r)_x \\ &= ((p^s + p^r)\alpha - p^r)_x \\ &= (p^s + p^r)\alpha_x + (p_x^s + p_x^r)\alpha - p_x^r. \end{aligned} \tag{4.14}$$

The subscript x in the above equations and the following equations denotes that the function is differentiated with respect to x .

4.3.2 Mathematical Formulation

To summarize the above constraints, the following two expressions are to be minimized simultaneously:

$$\int_C f^2 dx$$

$$\int_C g^2 dx$$

where f and g are as given in Eqs. 4.13 and 4.14. respectively. In order to optimally minimize the above two integrals simultaneously, a new function G is created

$$G = f^2 + \lambda^2 g^2$$

and the calculus of variations technique is applied to the integral

$$\int_C G dx.$$

The new function G explicitly depends upon the independent variable x , $\alpha(x)$, and its derivative $d\alpha(x)/dx$. The constant λ^2 must be regarded as a weight factor which allows for emphasis of either of the principles mentioned above.

If there exists a function $\alpha = \alpha(x)$ which minimizes the integral

$$\int_C G(x, \alpha, \alpha_x) dx,$$

then it must satisfy the Euler-Lagrange equation [34]:

$$\frac{\partial G}{\partial \alpha} - \frac{d}{dx} \left(\frac{\partial G}{\partial \alpha_x} \right) = 0$$

with

$$\begin{aligned} G &= f^2 + \lambda^2 g^2 \\ &= (1 - p^r - (p^s - p^r)\alpha)^2 + \\ &\quad \lambda^2 [(p^s + p^r)\alpha_x + (p_x^s + p_x^r)\alpha - p_x^r]^2. \end{aligned}$$

The Euler-Lagrange equation is calculated as follows:

$$\frac{\partial G}{\partial \alpha} = 2f(p^r - p^s) + 2\lambda^2 g(p_x^s + p_x^r)$$

$$\frac{d}{dx} \left(\frac{\partial G}{\partial \alpha_x} \right) = 2\lambda^2 g(p_x^s + p_x^r) + 2\lambda^2 (p^s + p^r) g_x$$

$$\begin{aligned} \frac{dg}{dx} = & (p_x^s + p_x^r) \alpha_x + (p^s + p^r) \alpha_{xx} + \\ & (p_{xx}^s + p_{xx}^r) \alpha + (p_x^s + p_x^r) \alpha_x - p_{xx}^r. \end{aligned}$$

Finally, by substituting the expressions of f and of dg/dx and by grouping the terms as coefficients of α and its partial derivative, the Euler-Lagrange equation can be written as:

$$\begin{aligned} \alpha_{xx} + \left[\frac{p_{xx}^s + p_{xx}^r}{p^s + p^r} - \left(\frac{p^s - p^r}{\lambda(p^s + p^r)} \right)^2 \right] \alpha \\ + \frac{2(p_x^s + p_x^r)}{p^s + p^r} \alpha_x = \frac{(1 - p^r)(p^r - p^s)}{\lambda^2(p^s + p^r)^2} + \frac{p_{xx}^r}{p^s + p^r}. \end{aligned}$$

In summary, the first order approximation of $\mathcal{F}$

$$\alpha_{xx} + A_1 \alpha_x + A_2 \alpha = C \quad (4.15)$$

when modeled continuously, is equivalent to solving one partial differential equation in α of the form shown by the above expression.

4.4 Numerical Solution

The original problem was the fusion of discrete stereo and range data. In order to ease the formulation of the fusion algorithm, these measurements were modelled by continuous density functions. By approximating the fusion functional by the first order truncation of its Taylor series, the problem of fusing two

density functions was reduced to one of determining the continuous function $\alpha(x)$ in the equation:

$$p(x) = \alpha(x)p^s(x) + (1 - \alpha(x))p^r(x)$$

It was shown in the previous section that, by means of imposing several constraints and applying the calculus of variations technique, $\alpha(x)$ must be a solution of the differential equation:

$$\frac{d^2\alpha(x)}{dx^2} + A_1(x)\frac{d\alpha(x)}{dx} + A_2\alpha(x) = C(x)$$

with

$$A_1(x) = 2\frac{p_x^s(x) + p_x^r(x)}{p^s(x) + p^r(x)}$$

$$A_2(x) = \left(\frac{p^s(x) - p^r(x)}{\lambda(p^s(x) + p^r(x))} \right)^2 + \frac{p_{xx}^s(x) + p_{xx}^r(x)}{p^s(x) + p^r(x)}$$

$$C(x) = \left(\frac{(1 - p^r(x))(p^s(x) - p^r(x))}{\lambda^2(p^s(x) + p^r(x))^2} \right) + \frac{p_{xx}^r(x)}{p^s(x) + p^r(x)}.$$

Returning to the discrete domain with the new notation $\alpha(i)$, $A_1(i)$, $A_2(i)$, and $C(i)$ to emphasize the discrete character of the following equations the transformation from the continuous domain to the discrete domain is obtained by means of the following approximations:

$$\alpha(x) = \alpha(i)$$

$$\alpha_x(x) = (\alpha(i+1) - \alpha(i-1))/2h$$

$$\alpha_{xx}(x) = (\alpha(i+1) - 2\alpha(i) + \alpha(i-1))/h^2.$$

By introducing these discrete approximations, the continuous differential equation becomes a difference equation of the form:

$$\alpha(i+1) \left(\frac{1}{h^2} + \frac{A_1(i+1)}{2h} \right) + \alpha(i) \left(\frac{-1}{h^2} + A_2(i) \right) + \alpha(i-1) \left(\frac{1}{h^2} - \frac{A_1(i-1)}{2h} \right) = C(i)$$

The coefficients are also computed discretely, and the derivatives which appear in their expressions are approximated in the very same manner as the derivatives in $\alpha(x)$:

$$A_1(i) = \frac{p^s(i+1) - p^s(i-1) + p^r(i+1) - p^r(i-1)}{(p^s(i) + p^r(i))}$$

$$A_2(i) = \left(\frac{p^s(i) - p^r(i)}{\lambda(p^s(i) + p^r(i))} \right)^2 + \frac{p_{xx}^s(i) + p_{xx}^r(i)}{p^s(i) + p^r(i)}$$

$$C(i) = \frac{(1 - p^r(i))(p^s(i) - p^r(i))}{\lambda^2(p^s(i) + p^r(i))^2} + \frac{p_{xx}^r(i)}{p^s(i) + p^r(i)}$$

The difference equation derived above can be rewritten for the sake of conciseness as follows:

$$\alpha(i+1)a_i + \alpha(i)d_i + \alpha(i-1)b_i = C(i)$$

with

$$a_i = \left(\frac{1}{h^2} + \frac{A_1(i+1)}{2h} \right)$$

$$d_i = \left(\frac{-2}{h^2} + A_2(i) \right)$$

$$b_i = \left(\frac{1}{h^2} - \frac{A_1(i-1)}{2h} \right).$$

This resulting system can be written in a matrix form as follows:

$$\mathbf{A}\alpha = \mathbf{C}.$$

The direct solution is straightforward and consists of inverting the matrix $\mathbf{A}$ in order to obtain the unknown vector α :

$$\alpha = \mathbf{A}^{-1}\mathbf{C} \quad (4.16)$$

where

$$\mathbf{A} = \begin{bmatrix} d_1 & a_1 & 0 & 0 & \cdots & \cdots & \cdots & 0 \\ b_2 & d_2 & a_2 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & b_3 & d_3 & a_3 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & b_4 & d_4 & a_4 & 0 & \cdots & 0 \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & 0 & 0 & b_{n-1} & d_{n-1} & a_{n-1} \\ 0 & 0 & 0 & 0 & 0 & 0 & b_n & d_n \end{bmatrix}.$$

$\mathbf{A}$ being a tridiagonal square matrix of size n , its inverse can be found easily.

The following section implements and illustrates the experimental results of the fusion algorithm.

4.5 Implementation and Experimental Results

This section illustrates by means of synthetic data and then by real data the previously described method for the fusion of two density functions.

As discussed in the previous section the mathematical expression which is minimized is $\mathcal{F}(\alpha) = (\text{corroboration}(\alpha) + \lambda \text{ validity}(\alpha))$. As we have no knowledge of how to choose or constrain λ , it is left as a parameter. Tests were conducted with several values of λ .

The α 's obtained should theoretically fall in the interval zero to one. However, because of numerical errors, some values of α are negative and some are greater than one. This result can be explained by the fact that the second derivatives of the density functions are required in the computation of α 's and the high noise sensitivity of the second derivatives yield erroneous results which affect values of α . A pure rescaling between zero and one would compress the sensible values of α in a narrow interval.

The Gaussian elimination with partial pivoting [35] method was implemented to factor a tridiagonal matrix and simultaneously solve a system of linear equations 4.16.

4.5.1 Synthetic Data

The synthetic data consisted of two input normal densities with the same variance but different means. The results are shown in Figs. 4.3 through 4.8. As

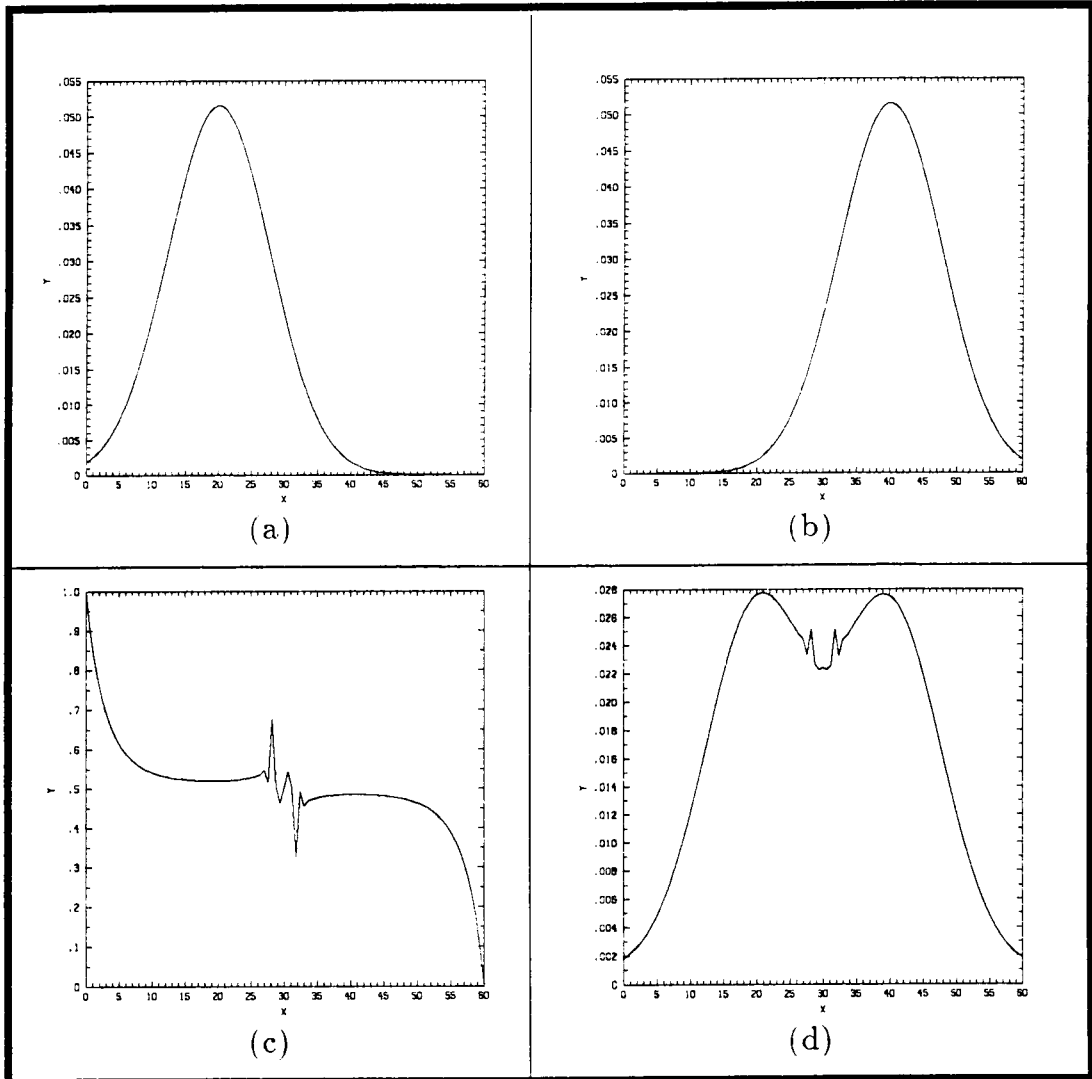


Figure 4.3: (a) normal input density 1 with mean = 20. (b) normal input density 2 with mean = 40, (c) Fusion function with $\lambda = 0.1$ (d) Fused density with mean = 29.94

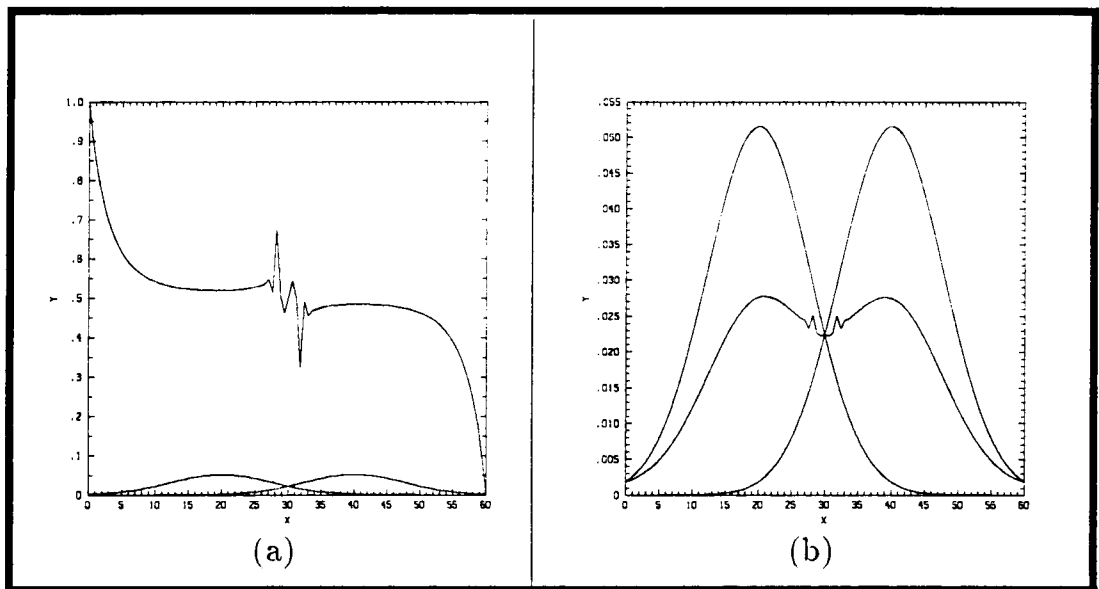


Figure 4.4: For $\lambda = 0.1$: (a) Fusion function superimposed on the two densities
(b) Fused density superimposed on the two densities

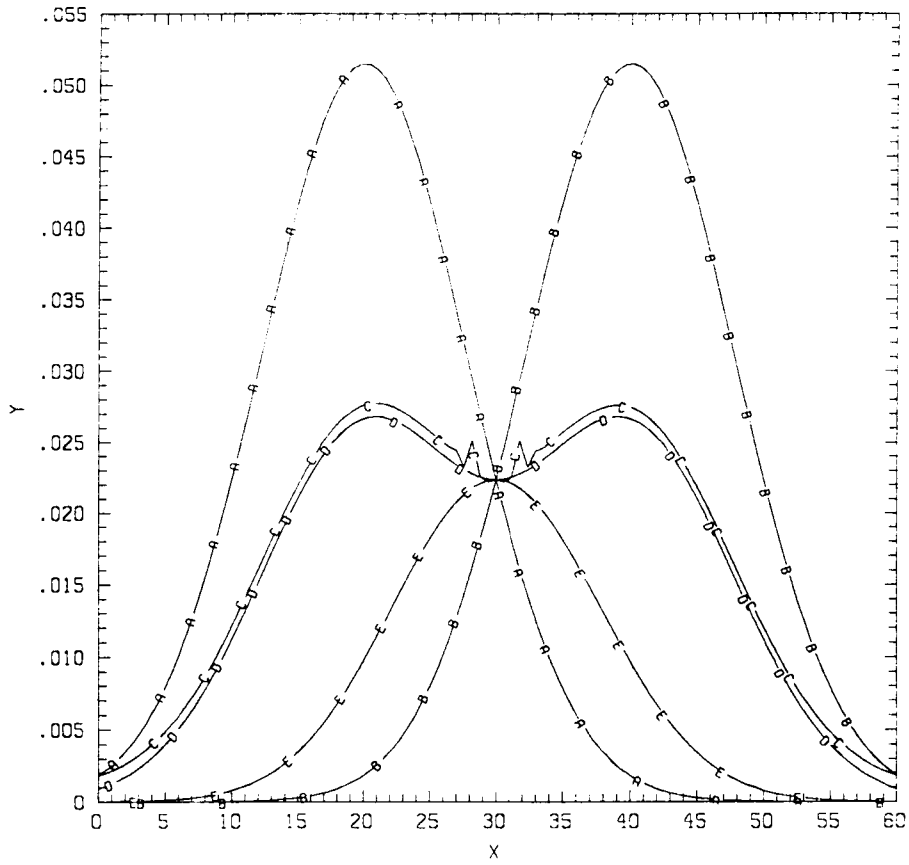


Figure 4.5: Fusion for $\lambda = 0.1$: A: Input density 1, B: Input density 2, C: Fusion of input densities, D: The arithmetic mean, E: The geometric mean

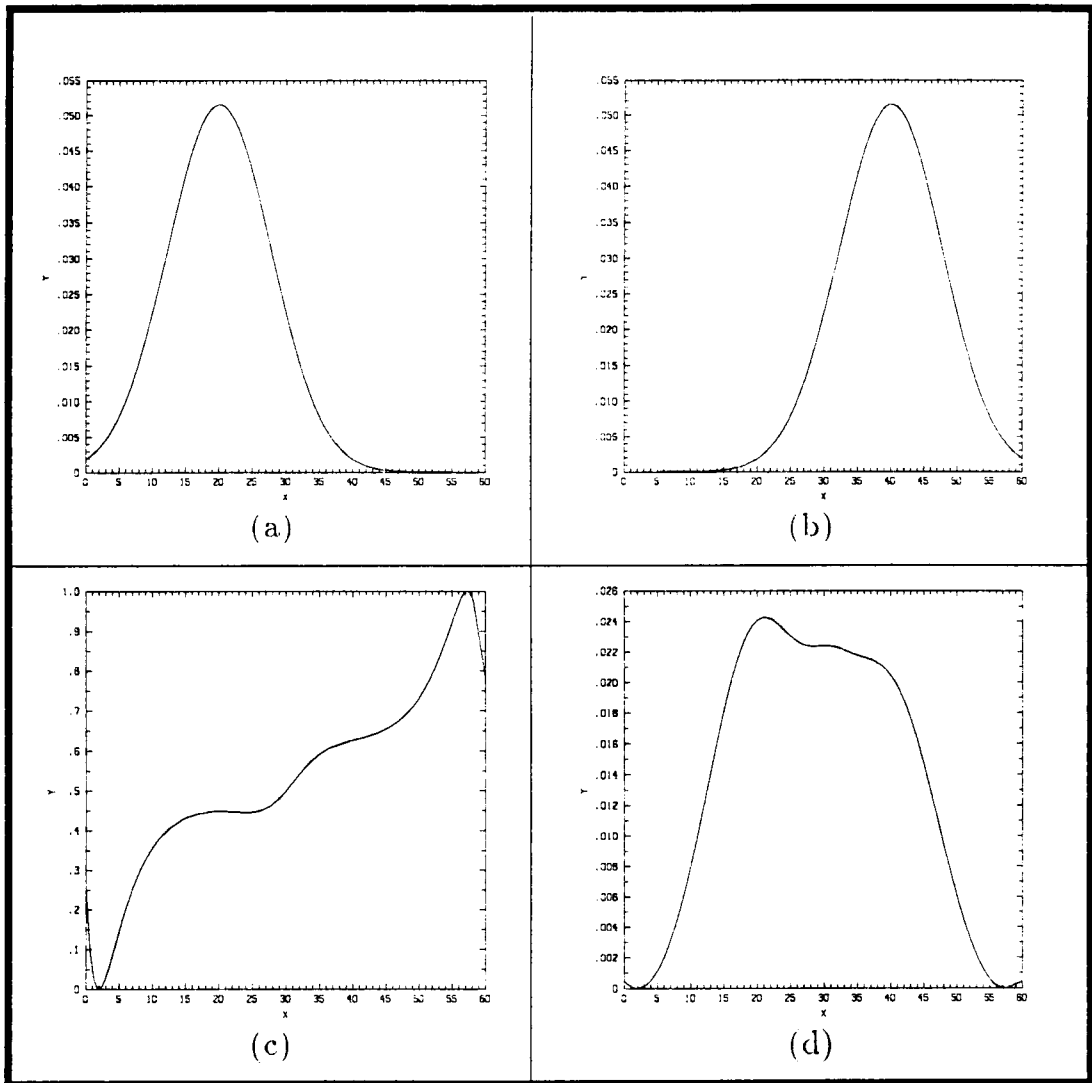


Figure 4.6: (a) normal input density 1 with mean = 20, (b) normal input density 2 with mean = 40, (c) Fusion function with $\lambda = 7.5$ (d) Fused density with mean 29.1

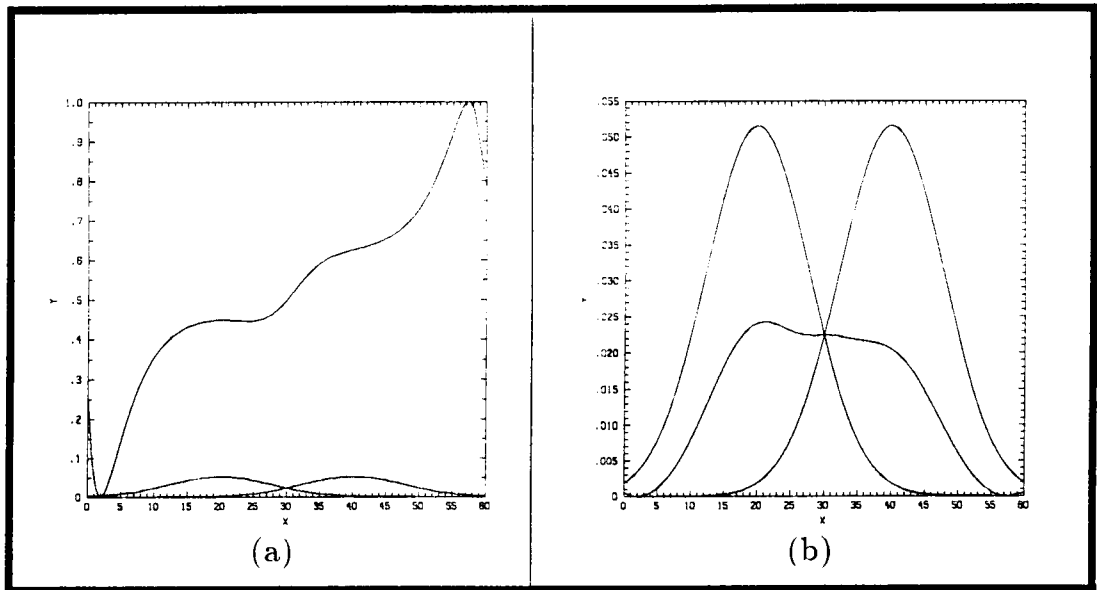


Figure 4.7: For $\lambda = 7.5$: (a) Fusion function superimposed on the two densities
 (b) Fused density superimposed on the two densities

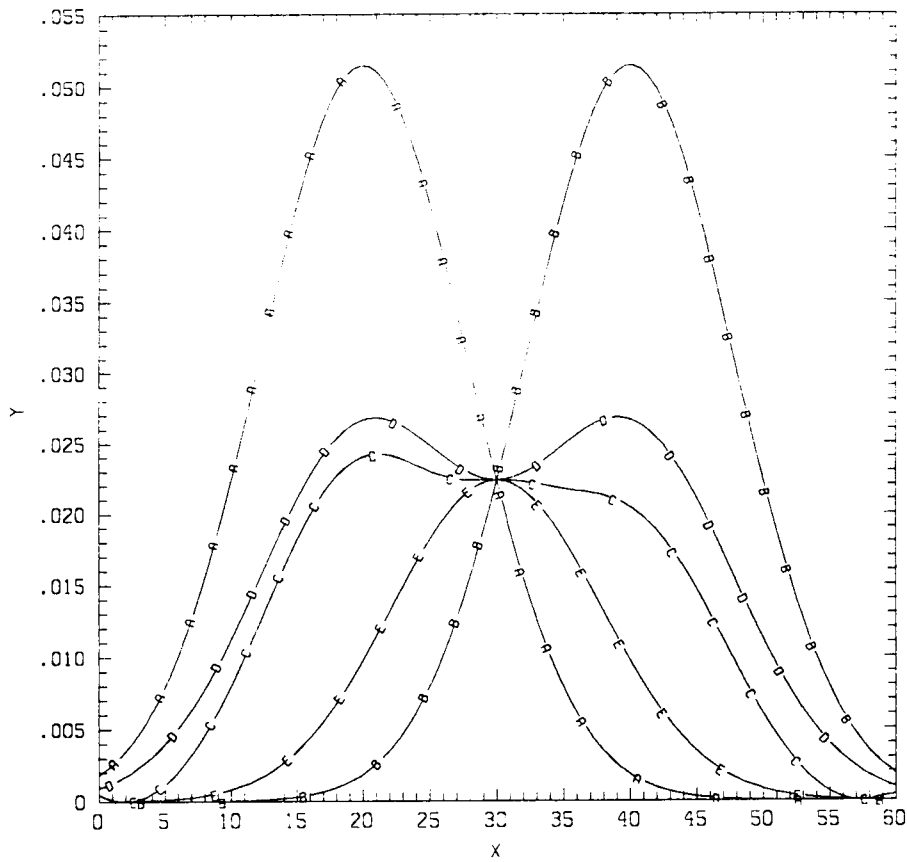


Figure 4.8: Fusion for $\lambda = 7.5$: A: Input density 1. B: Input density 2. C: Fusion of A and B. D: The arithmetic mean. E: The geometric mean

seen for low values of λ the fusion behaves as an arithmetic mean of the two densities, *i.e.*, the algorithm allows more sensor corroboration [Fig. 4.3]. Figure 4.4(b) shows a comparison of the fused density with the two input densities. Figure 4.5 illustrates the arithmetic and the geometric mean of the two input densities. As λ increases the fusion behaves as a geometric mean of the two densities, *i.e.*, the algorithm validates the two inputs (Figs. 4.6 and 4.8). Figure 4.7(b) shows the two input densities with the fused density. Figures 4.4(a) and 4.7(a) show the fusion function overlayed on the two input densities.

At this point it would be desirable to test the method on real intensity and range data.

4.5.2 Real Data

The input data consists of the intensity and range data modelled as probability density functions. Results of the fusion are provided. Detailed results are shown for the X coordinate of one 3-D point (on the quadrangular plane) only. The approach towards fusing the Y and Z coordinates is the same except for a change in the parameters. The results for the Y -, Z -coordinates of the first point and the 3-D coordinates of all the remaining three points of the object are illustrated finally.

Figure 4.9 shows the individual density functions of y_1 , y_2 , $y_1 - y_2$, and x_1 . The results are presented as follows. For each λ , three graphs are plotted on three consecutive pages. The first page illustrates the input densities to be fused

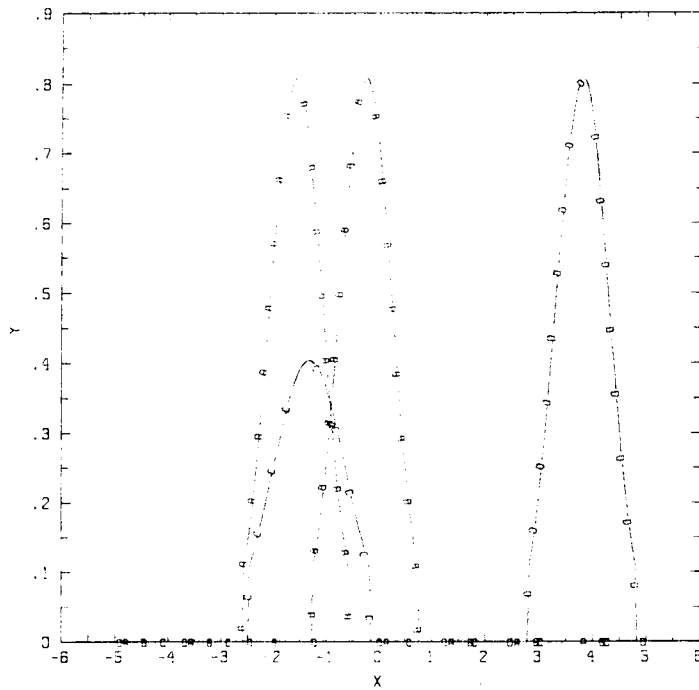


Figure 4.9: A represents the p.d.f $p(y_1)$, B represents the p.d.f $p(y_2)$, C represents the $p(y_1 - y_2)$, D represents the $p(x_1)$

in the top left and the top right corners of the pictures. The density containing the values of α , which is referred to as the fusion function, appears in the bottom left corner and the fused density ($\alpha \times \text{Input density 1} + (1 - \alpha) \times \text{Input density 2}$) appears in the right corner. The second page has two plots which show the fusion function and the fused density superimposed on the two input densities. The third page shows a comparison between the fused density, the arithmetic mean, and the geometric mean of the two input densities.

The following figures allow clear vision of the scope of variation of the algorithms results. For low values of λ , typically, $\lambda < 1$, the method favors the principle of sensor corroboration. This can be seen in Fig. 4.10 to Fig. 4.12. The area under the curve of input 1 between -30 and -26 is almost zero. Whereas there exists some area under the curve of input 2 which appears also under the fusion curve. So is the case between the points -4 and 0 . The fusion function $\alpha(i)$ takes the two extreme values of 0 and 1 at -30 and -13.5 respectively when integrating the two input densities. At the intersection of the two densities, the result behaves as an arithmetic mean of the two densities [Fig. 4.12]. At low values of λ , the fusion algorithm performs almost like a simple "OR" operation. It tends to select one density without confirming the other density.

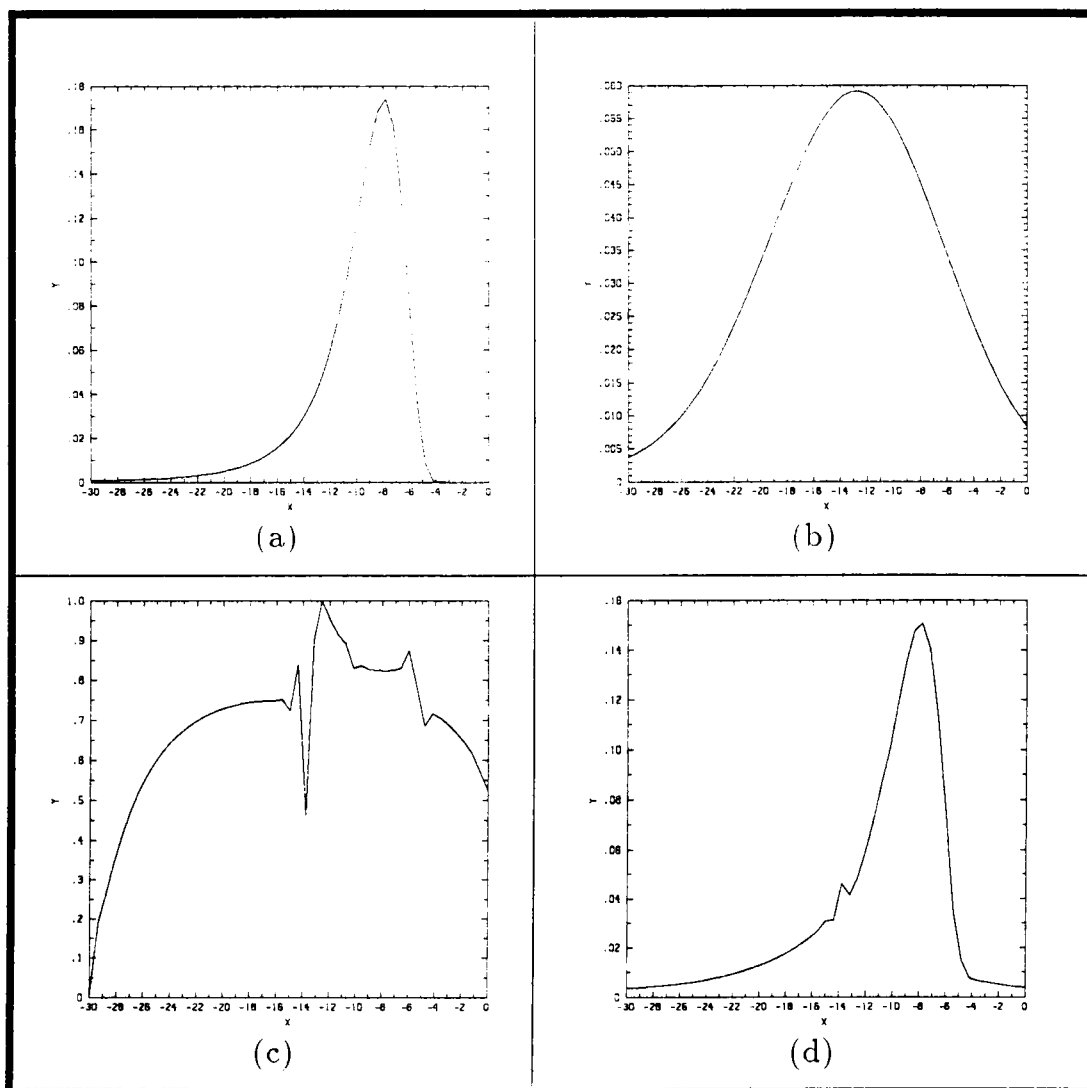


Figure 4.10: (a): p.d.f $p_{X_1}^s$ with mean $= -9.8$ and variance $= 11.8$. (b): p.d.f $p_{X_1}^r$ with mean $= -13.5$ and variance $= 38.2$. (c): Fusion function with $\lambda = 0.1$ (d): Fused density with mean -10.8 and variance 23.91

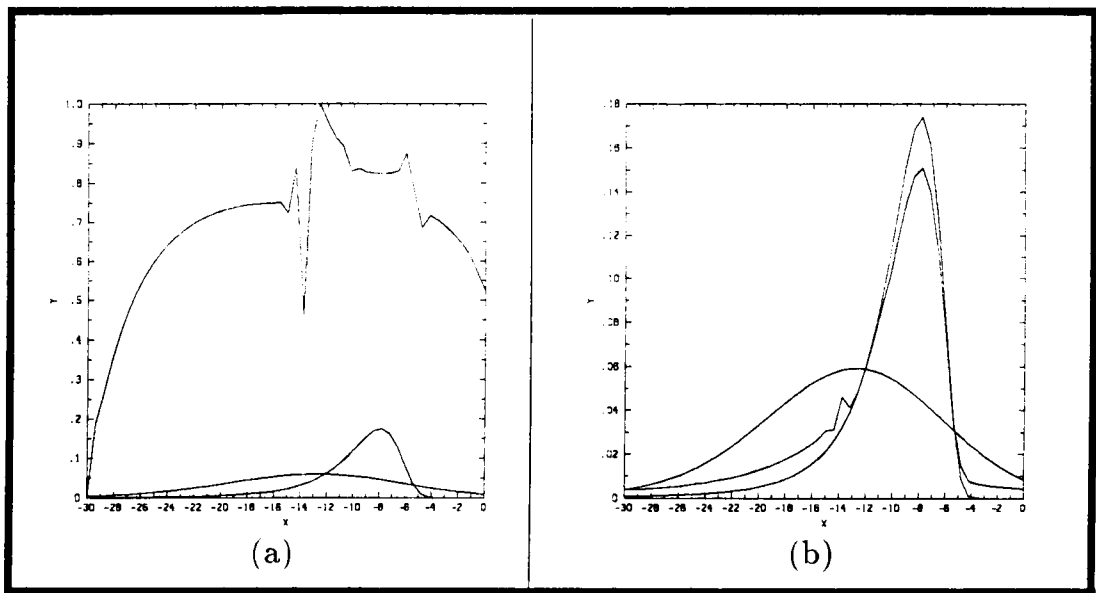


Figure 4.11: For $\lambda = 0.1$: (a) Fusion function superimposed on the two densities
(b) the fused density superimposed on the two densities

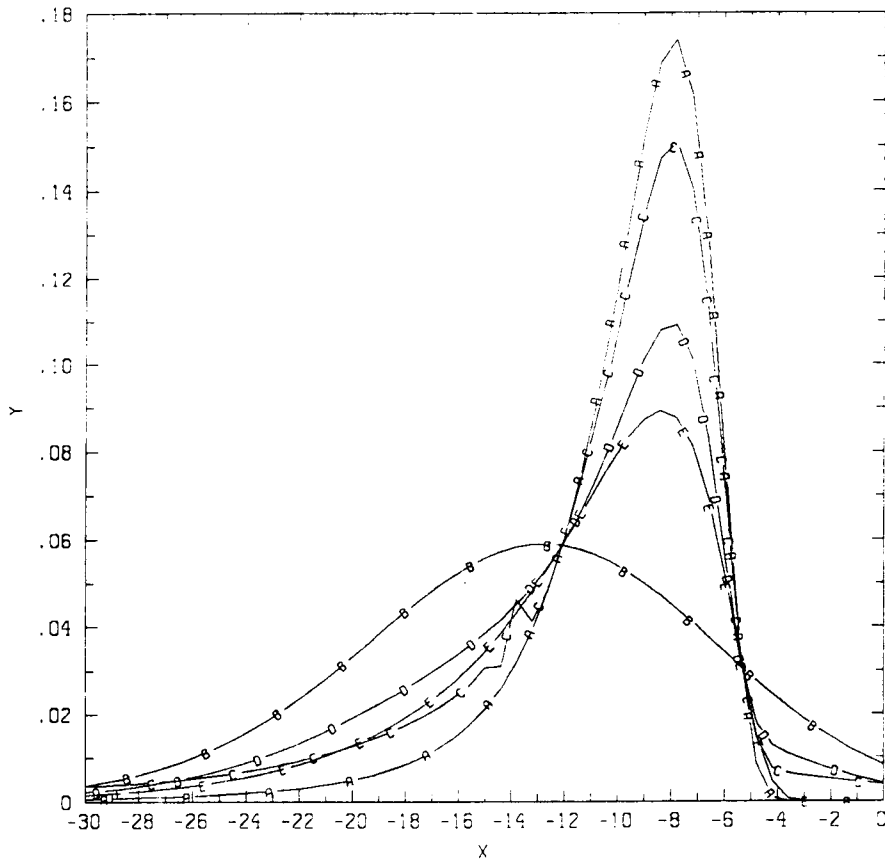


Figure 4.12: Fusion for $\lambda = 0.1$: A: Input density 1. B: Input density 2. C: Fusion of the two inputs. D: Arithmetic mean. E: Geometric mean

As λ increases, especially between the values of 0.25 and 0.75, the fusion function starts behaving as a critically damped system. This can be attributed to the inherent property of the solution of the second order differential equation which is typically very sensitive to noise. However the resulting density does not refute the principle of corroboration; this is evident between the region -4 to 0 in Figs. 4.13 to 4.21.

For large values of λ , typically for $\lambda > 1$, the method strongly emphasizes the principle of validity. The area under the input density functions that appear in both inputs appear in the resulting density. The result can be seen in Figs. 4.22 to 4.36. The input density function 1 goes to zero around -4 and the input density 2 if assumed to go to zero at 0 . It is clearly seen that the resultant density goes to zero around -2 . This small error can be attributed to rounding errors in the plotting and the computing routines. This trend of α going to the extremes is seen clearly as λ increases.

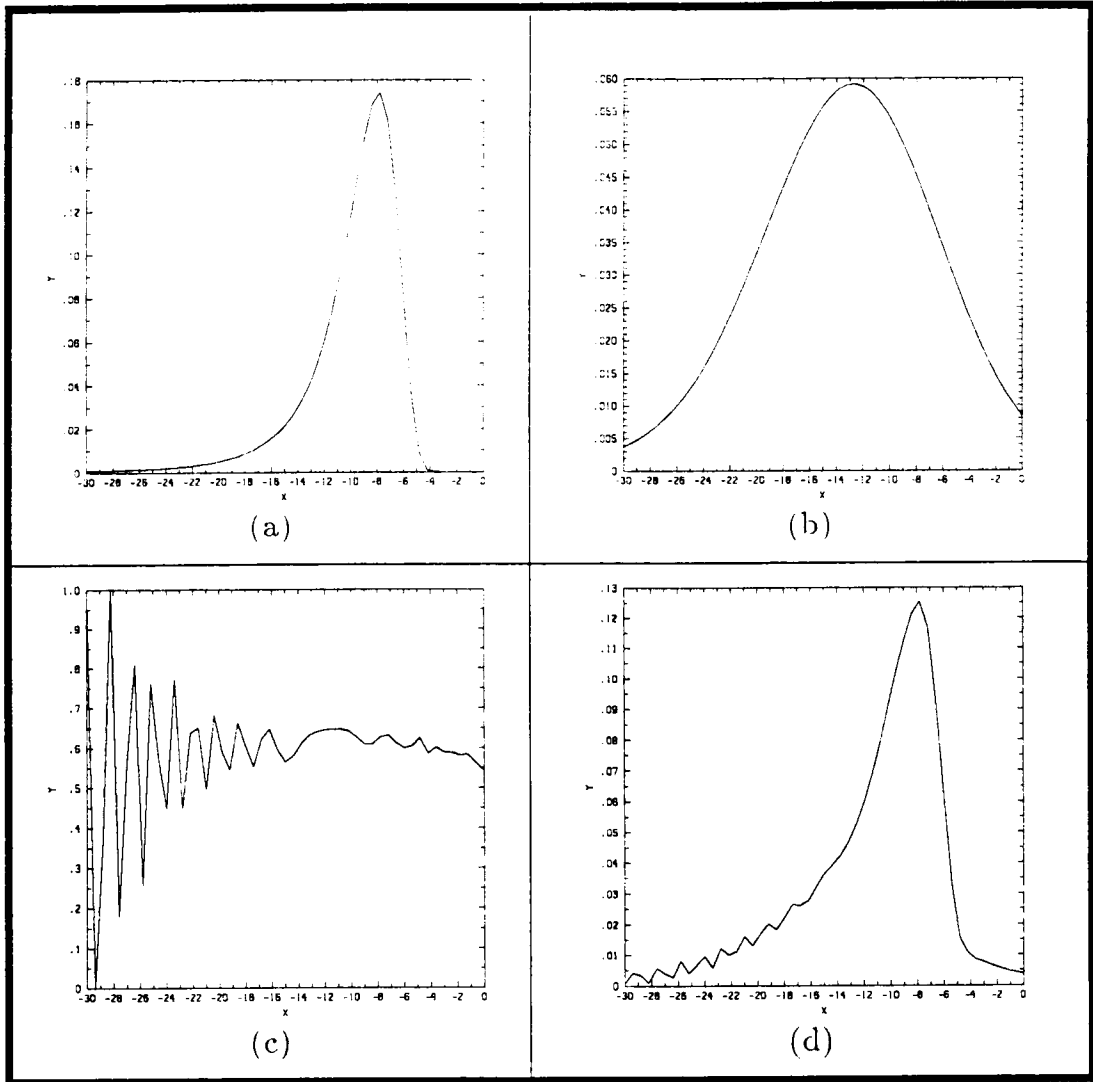


Figure 4.13: (a) p.d.f $p_{X_1}^s$ with mean = -9.8 and variance = 11.8 . (b): p.d.f $p_{X_1}^r$ with mean = -13.5 and variance = 38.2 . (c) Fusion function with $\lambda = 0.25$ (d) Fused density with mean -11.23 and variance 23.9

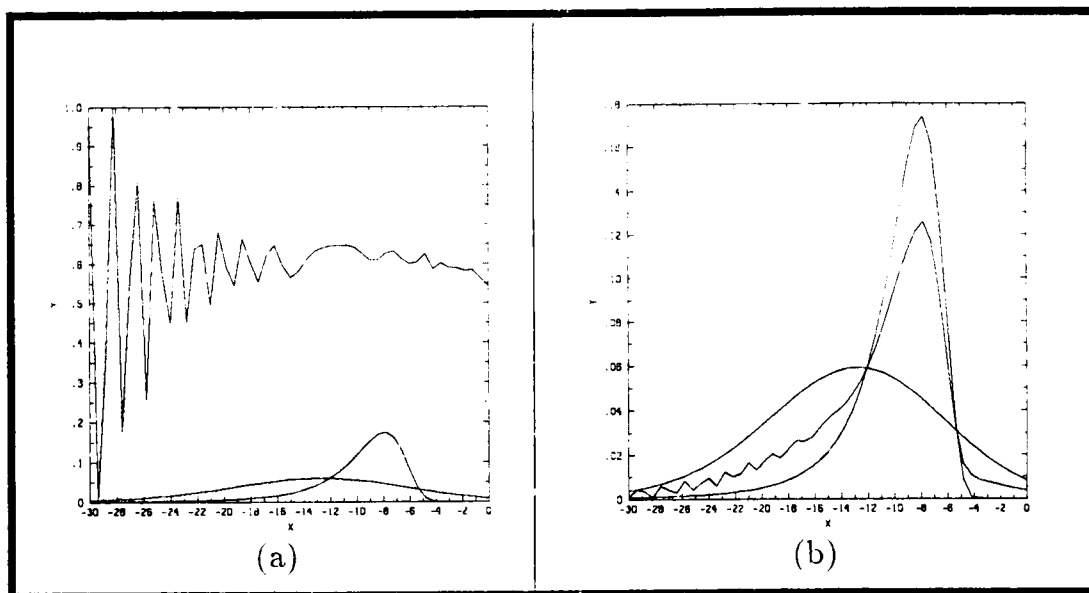


Figure 4.14: For $\lambda = 0.25$: (a) Fusion function superimposed on the two densities
(b) the fused density superimposed on the two densities

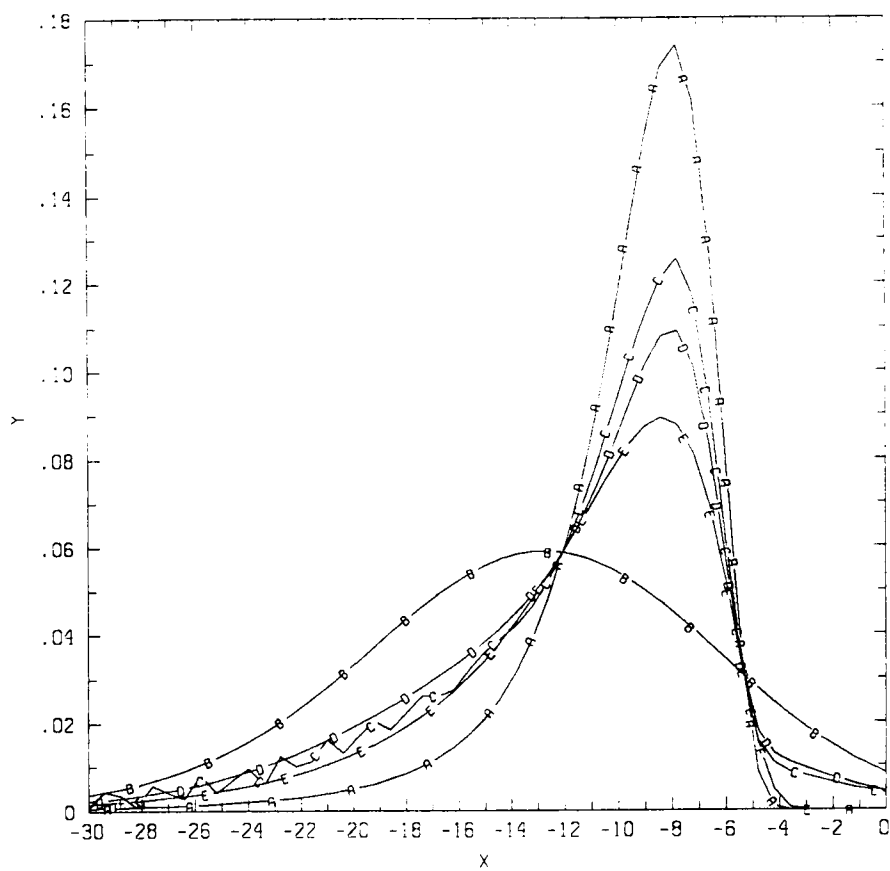


Figure 4.15: Fusion for $\lambda = 0.25$: A: Input density 1. B: Input density 2. C: Fusion of the two inputs. D: Arithmetic mean. E: Geometric mean

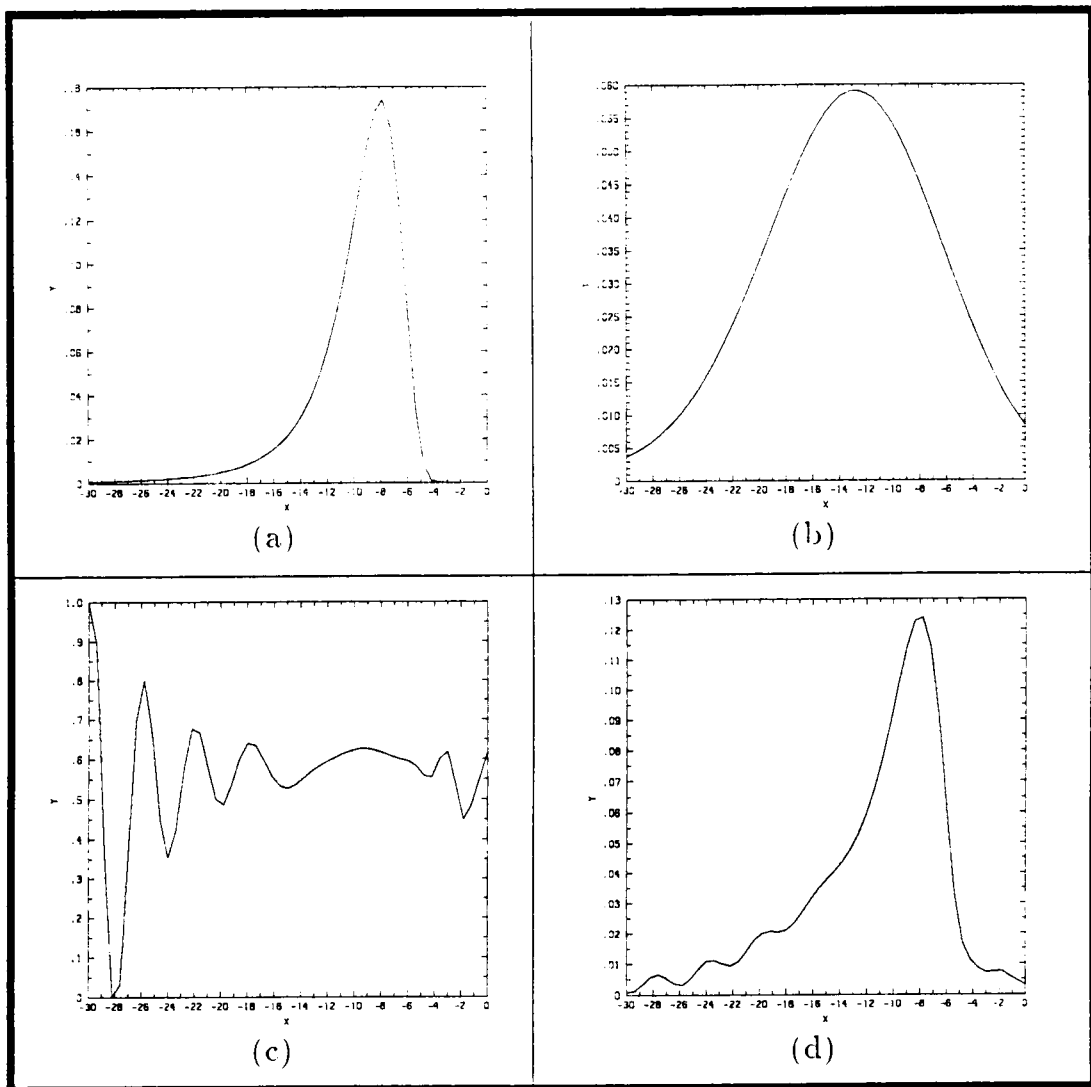


Figure 4.16: (a) p.d.f $p_{X_1}^s$ with mean = -9.8 and variance = 11.8 . (b): p.d.f $p_{X_1}^r$ with mean = -13.5 and variance = 38.2 . (c) Fusion function with $\lambda = 0.5$ (d) Fused density with mean -11.32 and variance 26.6

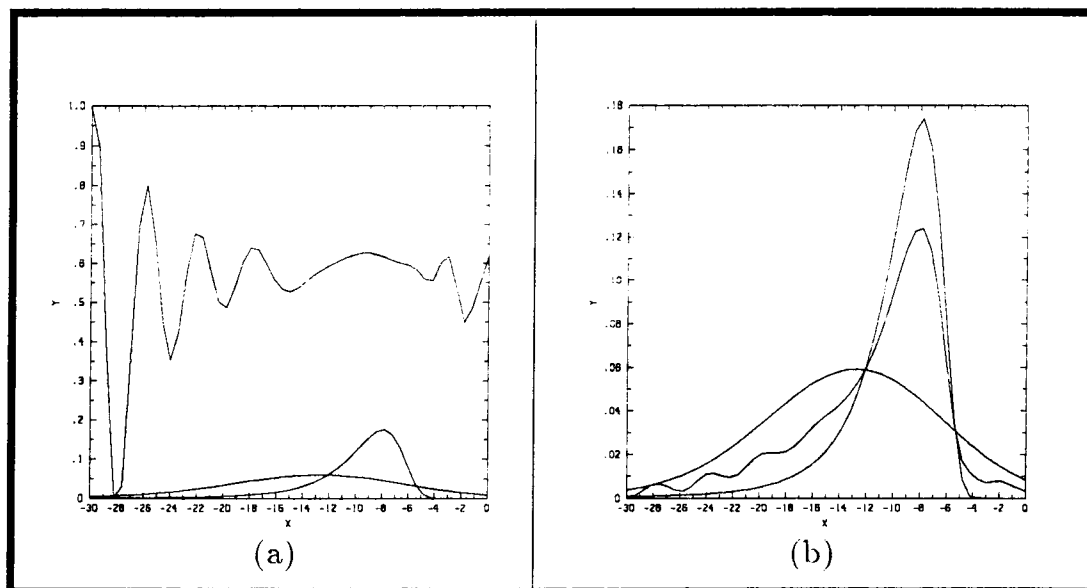


Figure 4.17: For $\lambda = 0.5$: (a) Fusion function superimposed on the two densities
(b) the fused density superimposed on the two densities

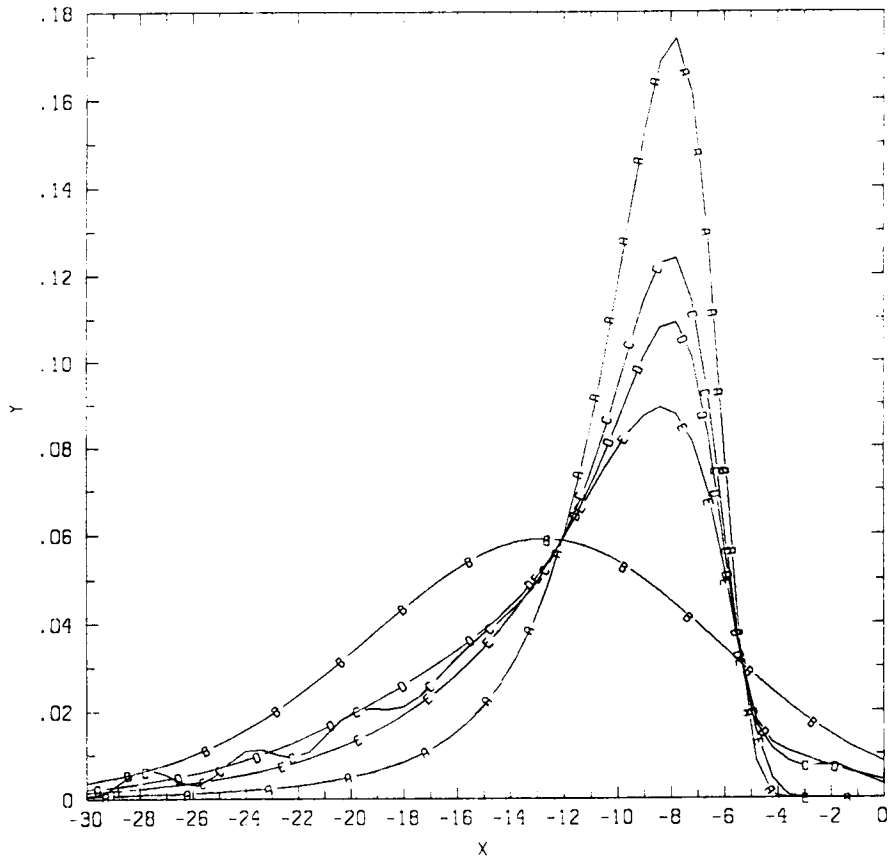


Figure 4.18: Fusion for $\lambda = 0.5$: A: Input density 1, B: Input density 2, C: Fusion of the two inputs, D: Arithmetic mean, E: Geometric mean

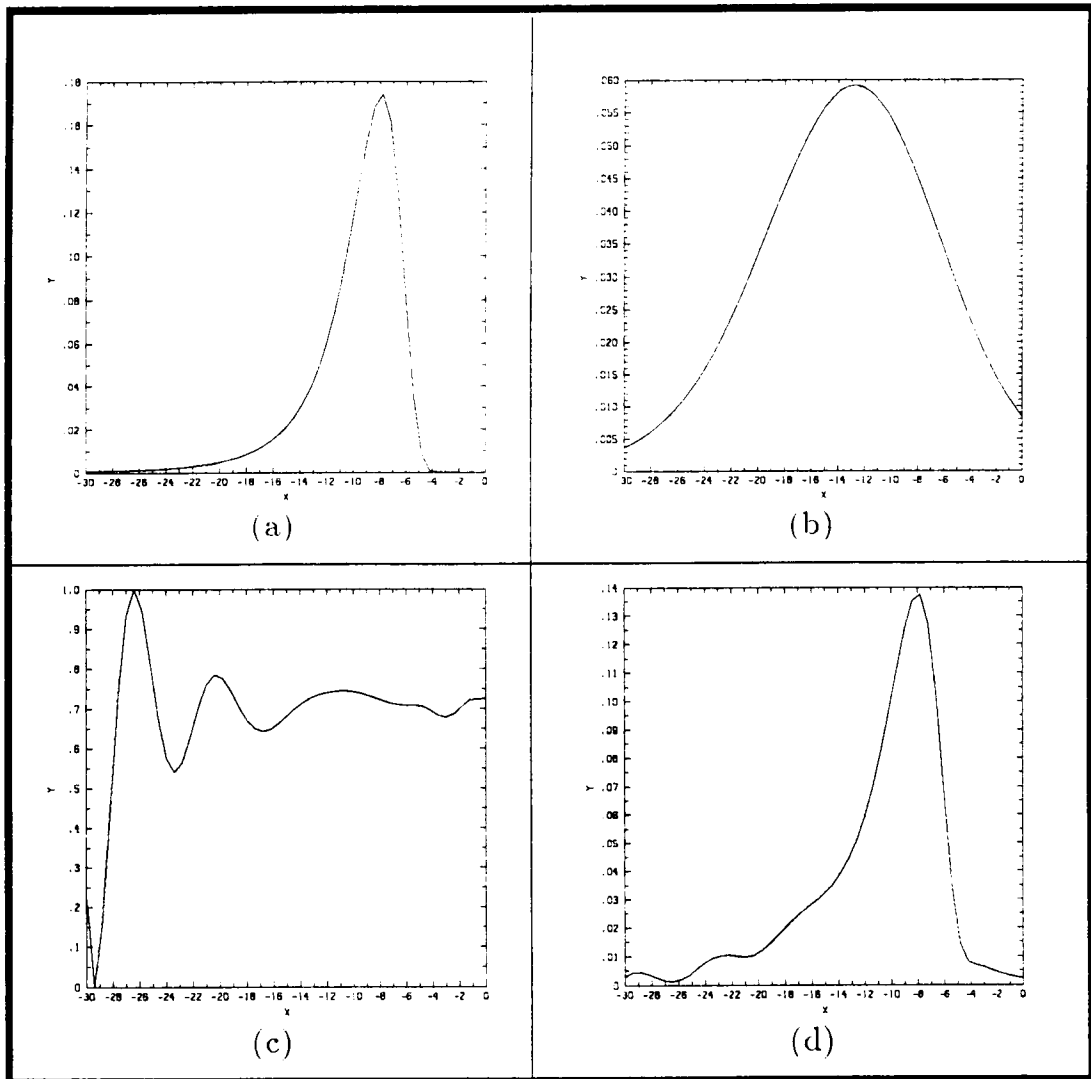


Figure 4.19: (a) p.d.f $p_{X_1}^s$ with mean $= -9.8$ and variance $= 11.8$. (b): p.d.f $p_{X_1}^r$ with mean $= -13.5$ and variance $= 38.2$. (c) Fusion function with $\lambda = 0.75$ (d) Fused density with mean -10.9 and variance 22.9

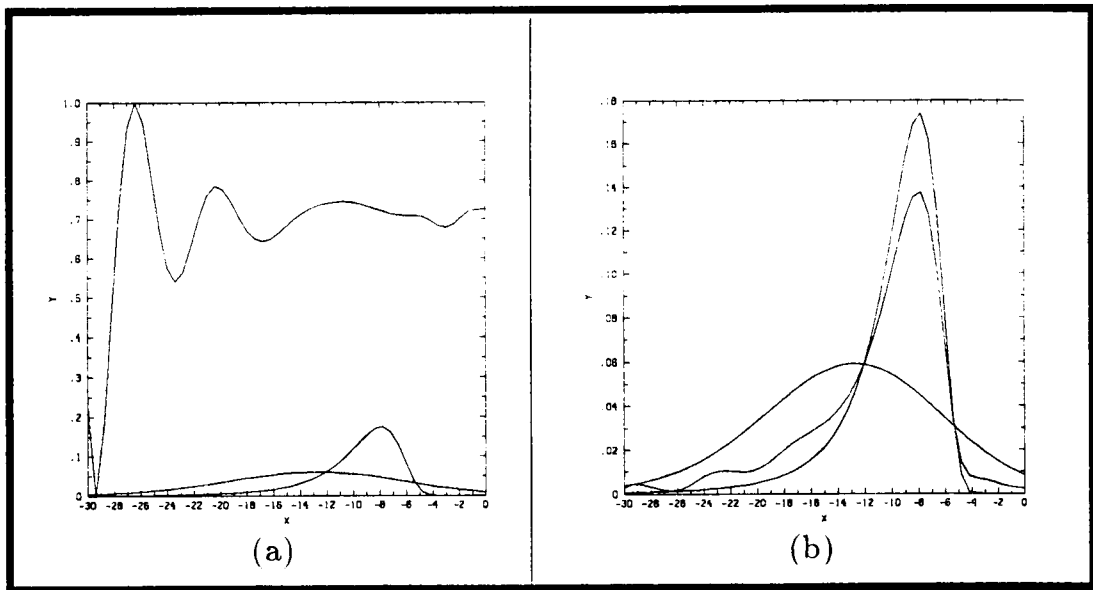


Figure 4.20: For $\lambda = 0.75$: (a) Fusion function superimposed on the two densities
(b) the fused density superimposed on the two densities

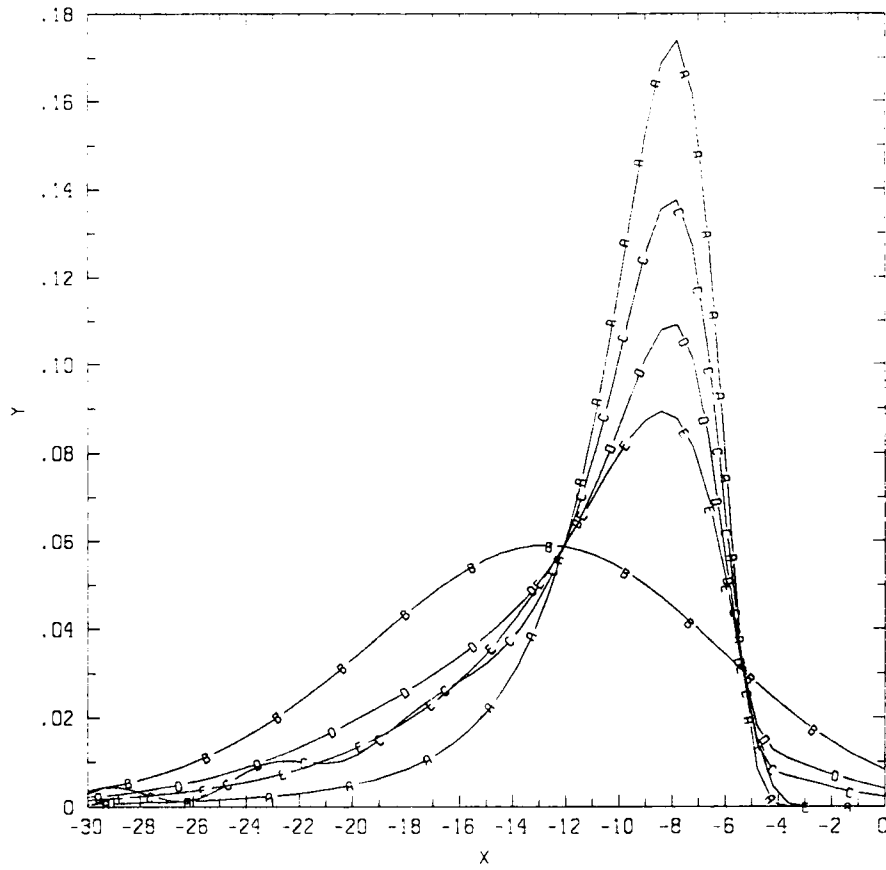


Figure 4.21: Fusion for $\lambda = 0.75$: A: Input density 1. B: Input density 2. C: Fusion of the two inputs. D: Arithmetic mean. E: Geometric mean

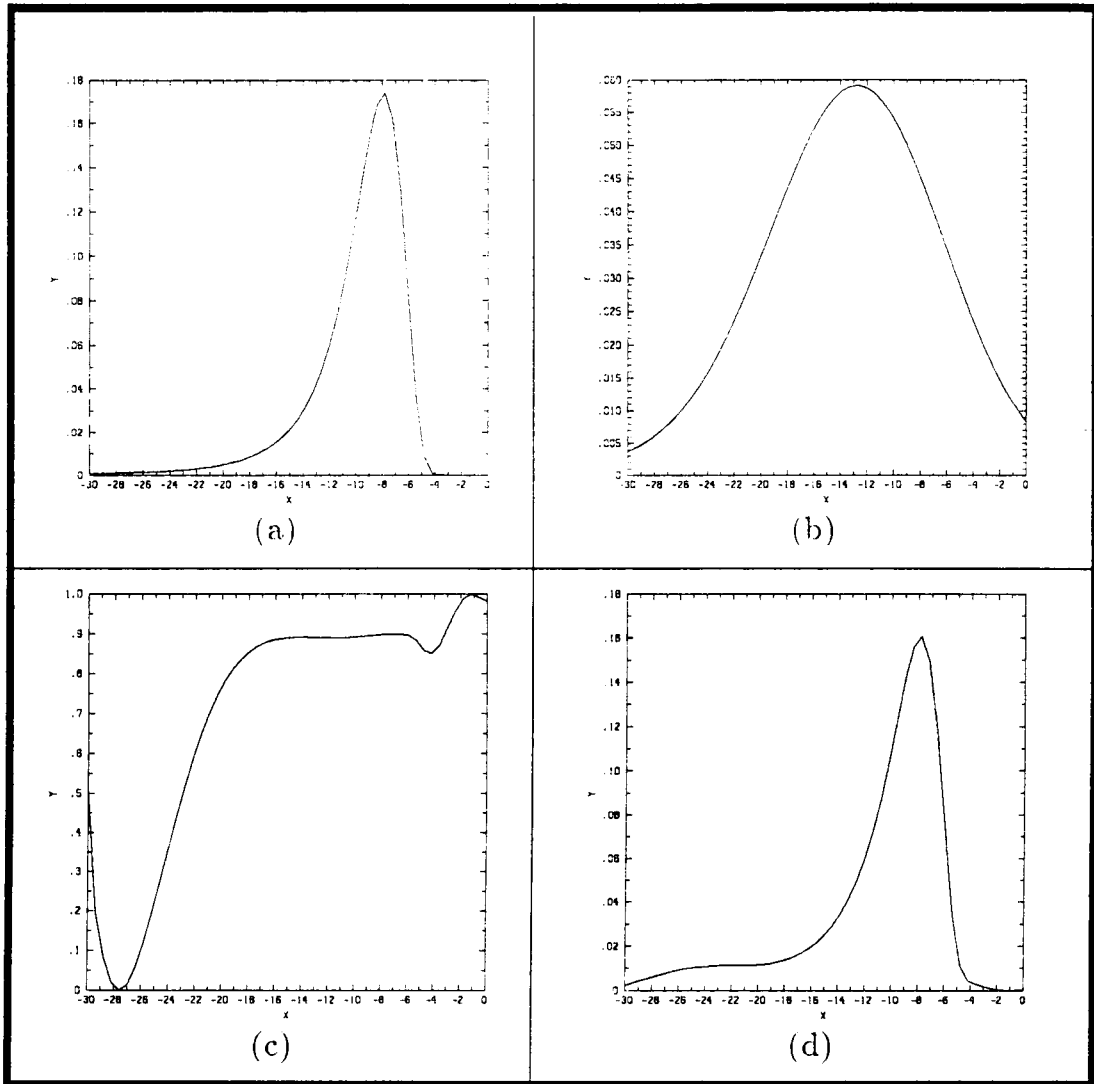


Figure 4.22: (a) p.d.f $p_{X_1}^s$ with mean = -9.8 and variance = 11.8 . (b): p.d.f $p_{X_1}^r$ with mean = -13.5 and variance = 38.2 . (c) Fusion function with $\lambda = 2.0$ (d) Fused density with mean -11.0 and variance 25.9

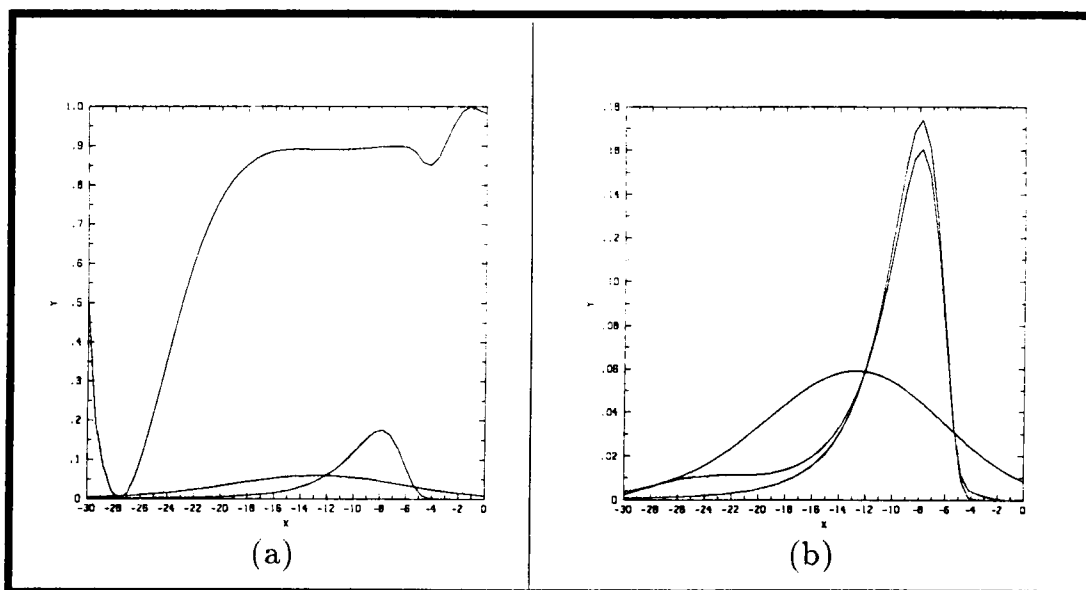


Figure 4.23: For $\lambda = 2.0$: (a) Fusion function superimposed on the two densities
(b) the fused density superimposed on the two densities

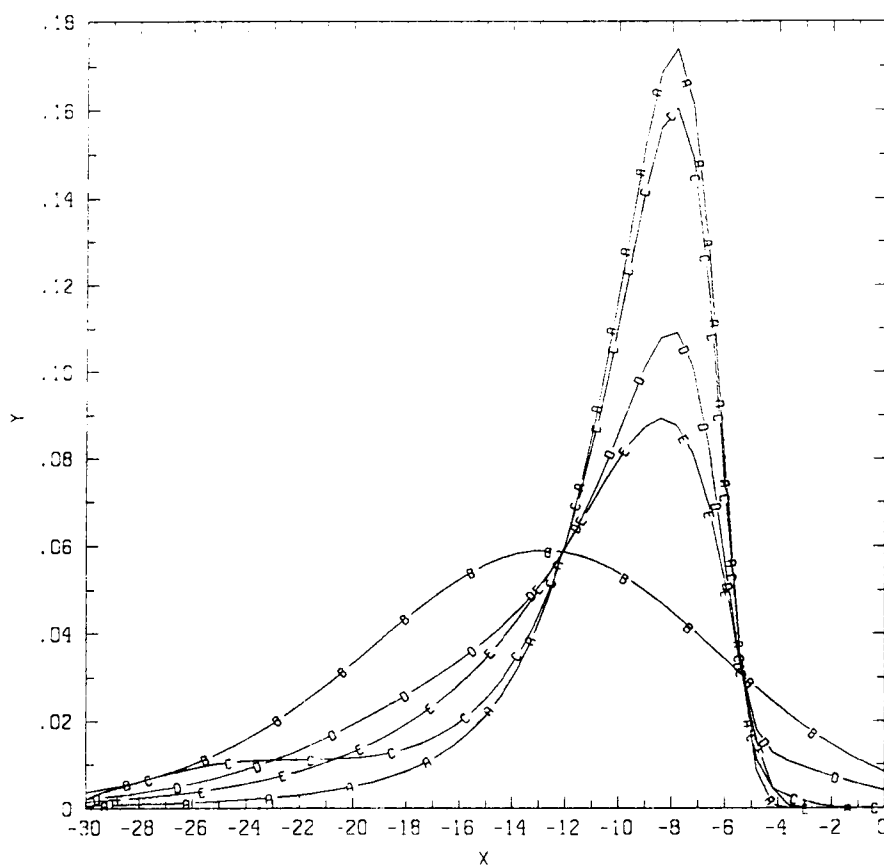


Figure 4.24: Fusion for $\lambda = 2.0$: A: Input density 1. B: Input density 2. C: Fusion of the two inputs. D: Arithmetic mean. E: Geometric mean

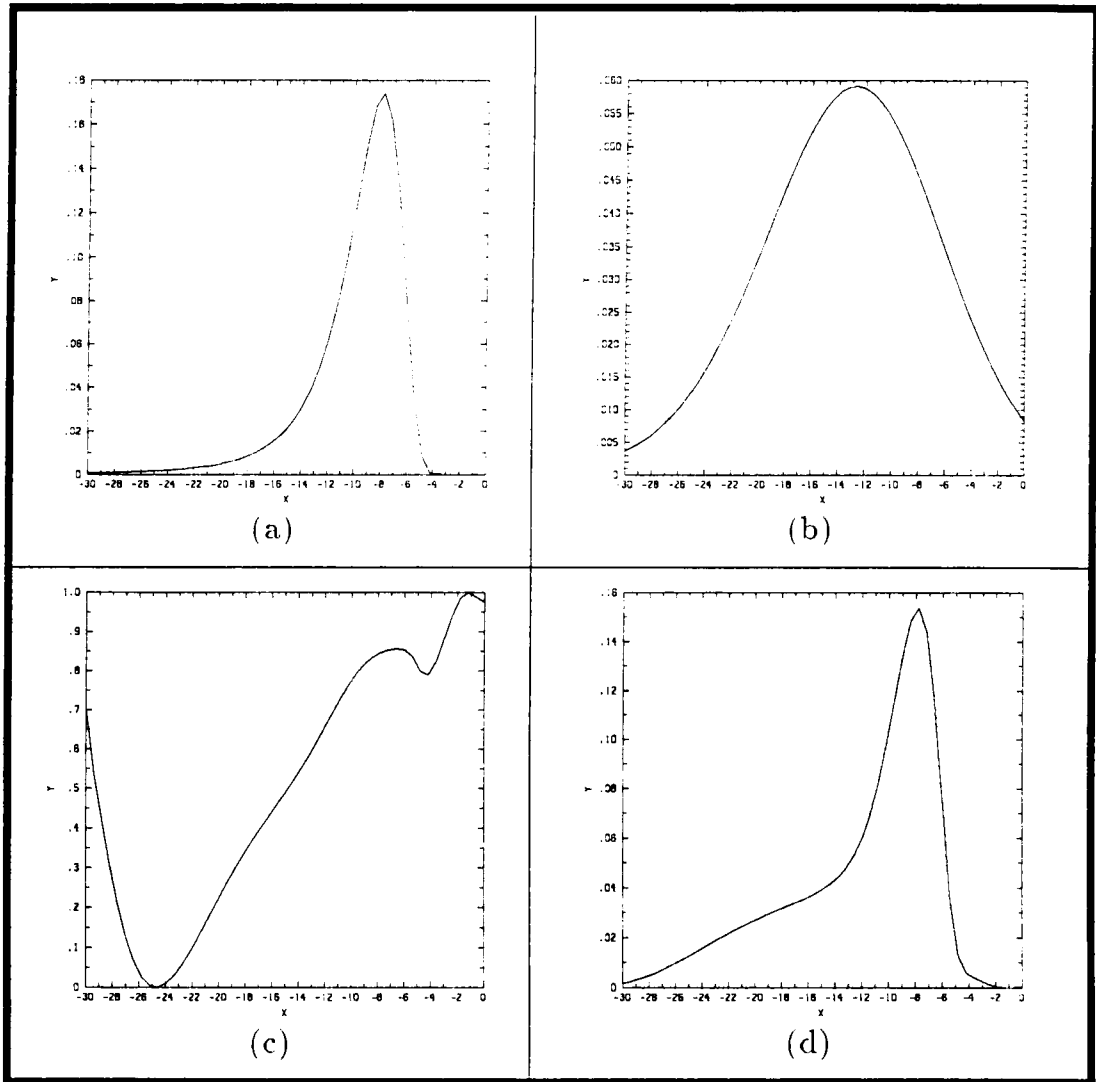


Figure 4.25: (a) p.d.f $p_{X_1}^s$ with mean = -9.8 and variance = 11.8 . (b): p.d.f $p_{X_1}^r$ with mean = -13.5 and variance = 38.2 . (c) Fusion function with $\lambda = 2.5$ (d) Fused density with mean -11.96 and variance 29.6

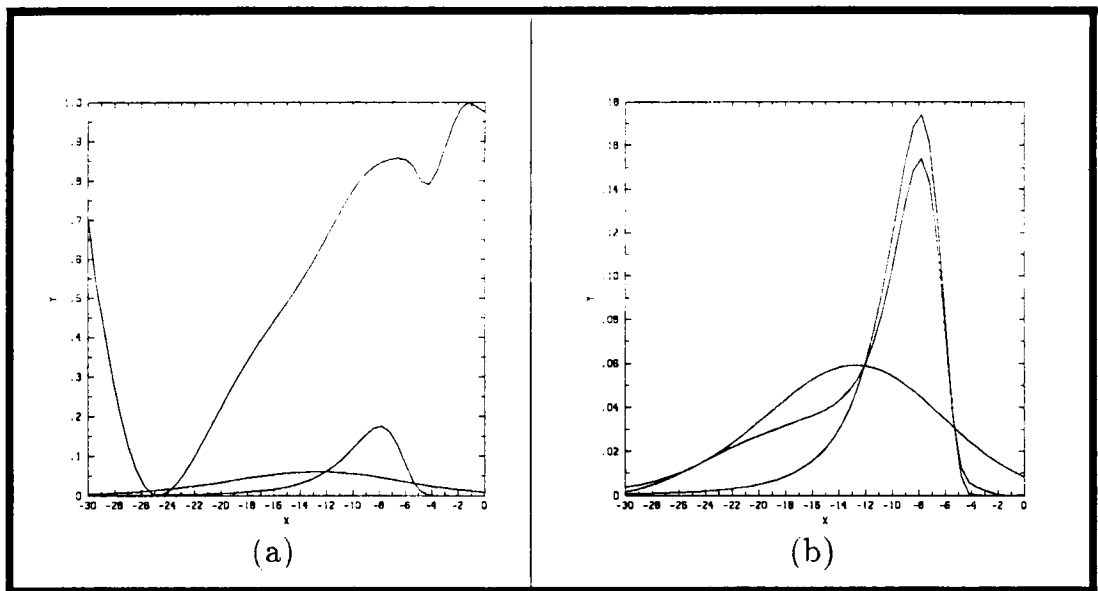


Figure 4.26: For $\lambda = 2.5$: (a) Fusion function superimposed on the two densities
 (b) the fused density superimposed on the two densities

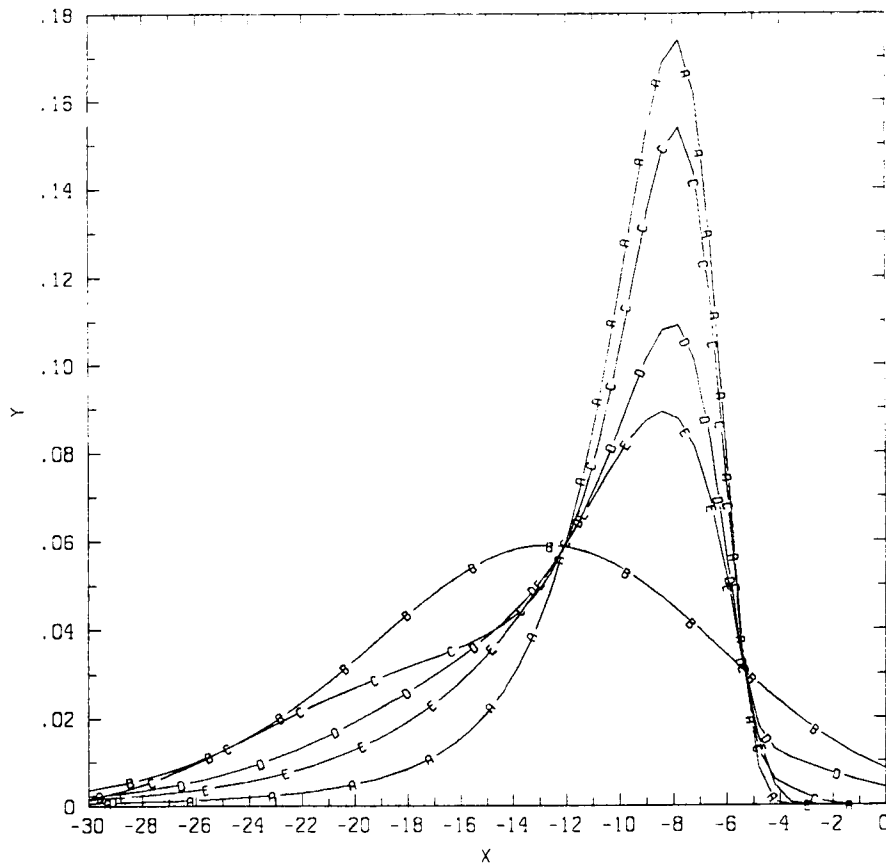


Figure 4.27: Fusion for $\lambda = 2.5$: A: Input density 1, B: Input density 2, C: Fusion of the two inputs, D: Arithmetic mean, E: Geometric mean

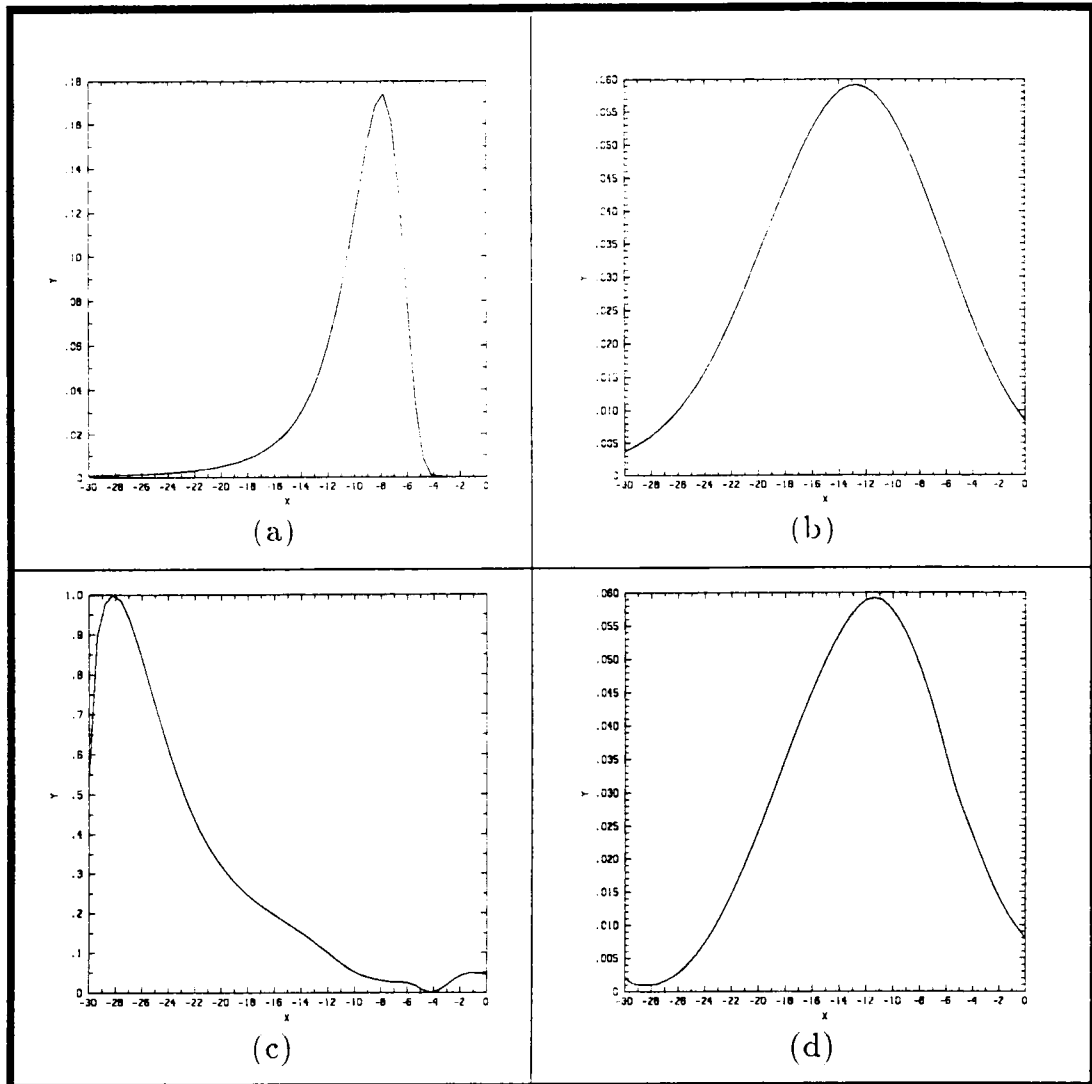


Figure 4.28: (a) p.d.f $p_{X_1}^s$ with mean $= -9.8$ and variance $= 11.8$. (b): p.d.f $p_{X_1}^r$ with mean $= -13.5$ and variance $= 38.2$. (c) Fusion function with $\lambda = 5.0$ (d) Fused density with mean -12.3 and variance 30.4

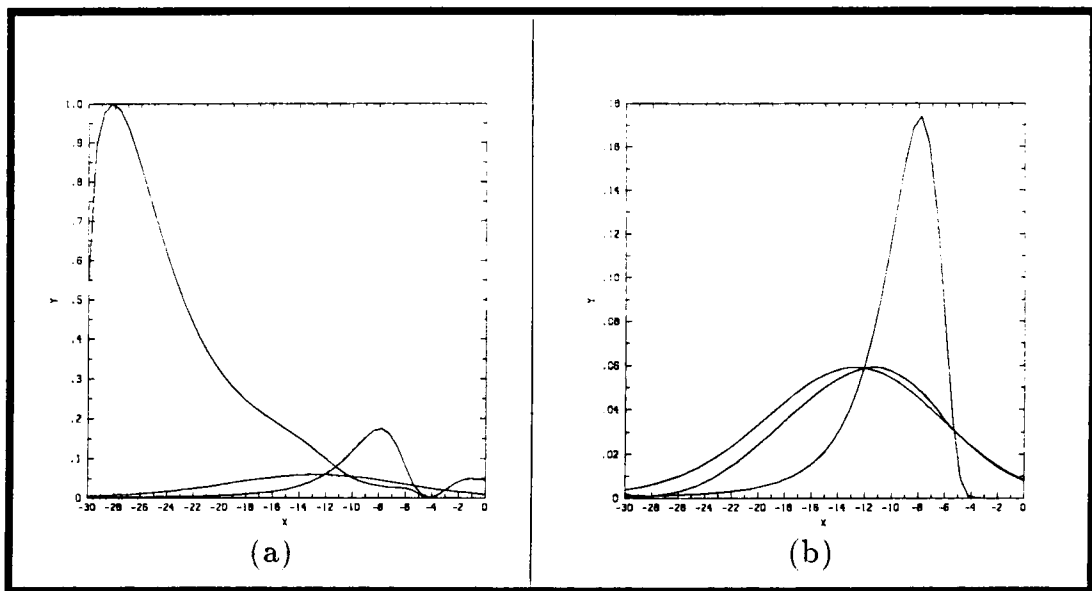


Figure 4.29: For $\lambda = 5.0$: (a) Fusion function superimposed on the two densities
(b) the fused density superimposed on the two densities

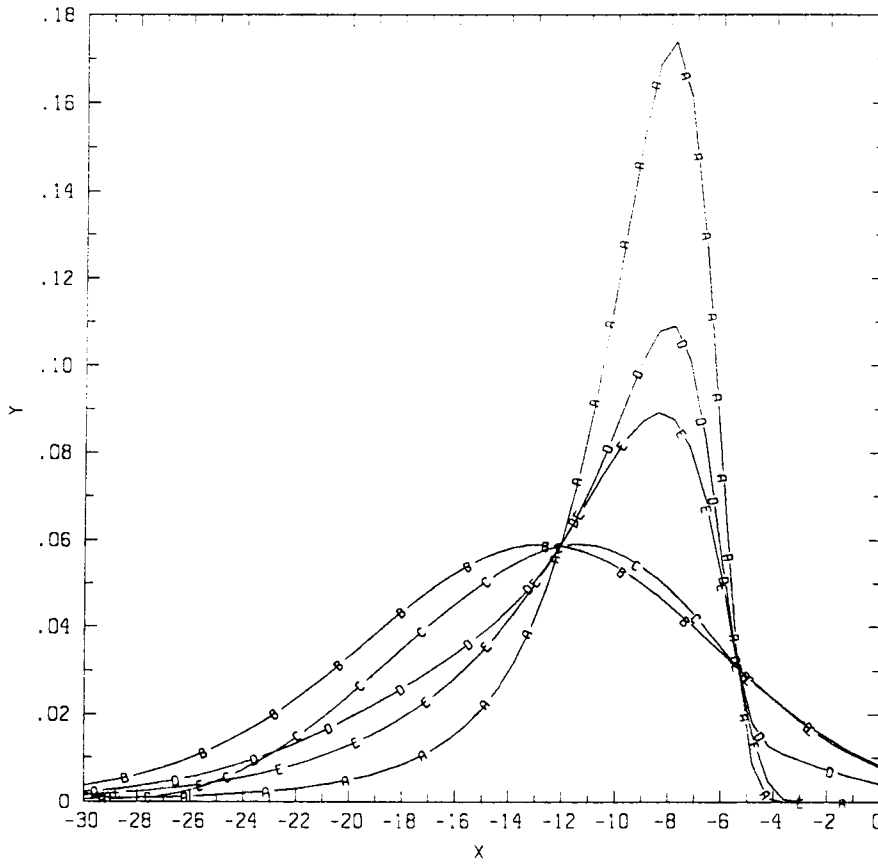


Figure 4.30: Fusion for $\lambda = 5.0$: A: Input density 1, B: Input density 2, C: Fusion of the two inputs, D: Arithmetic mean, E: Geometric mean

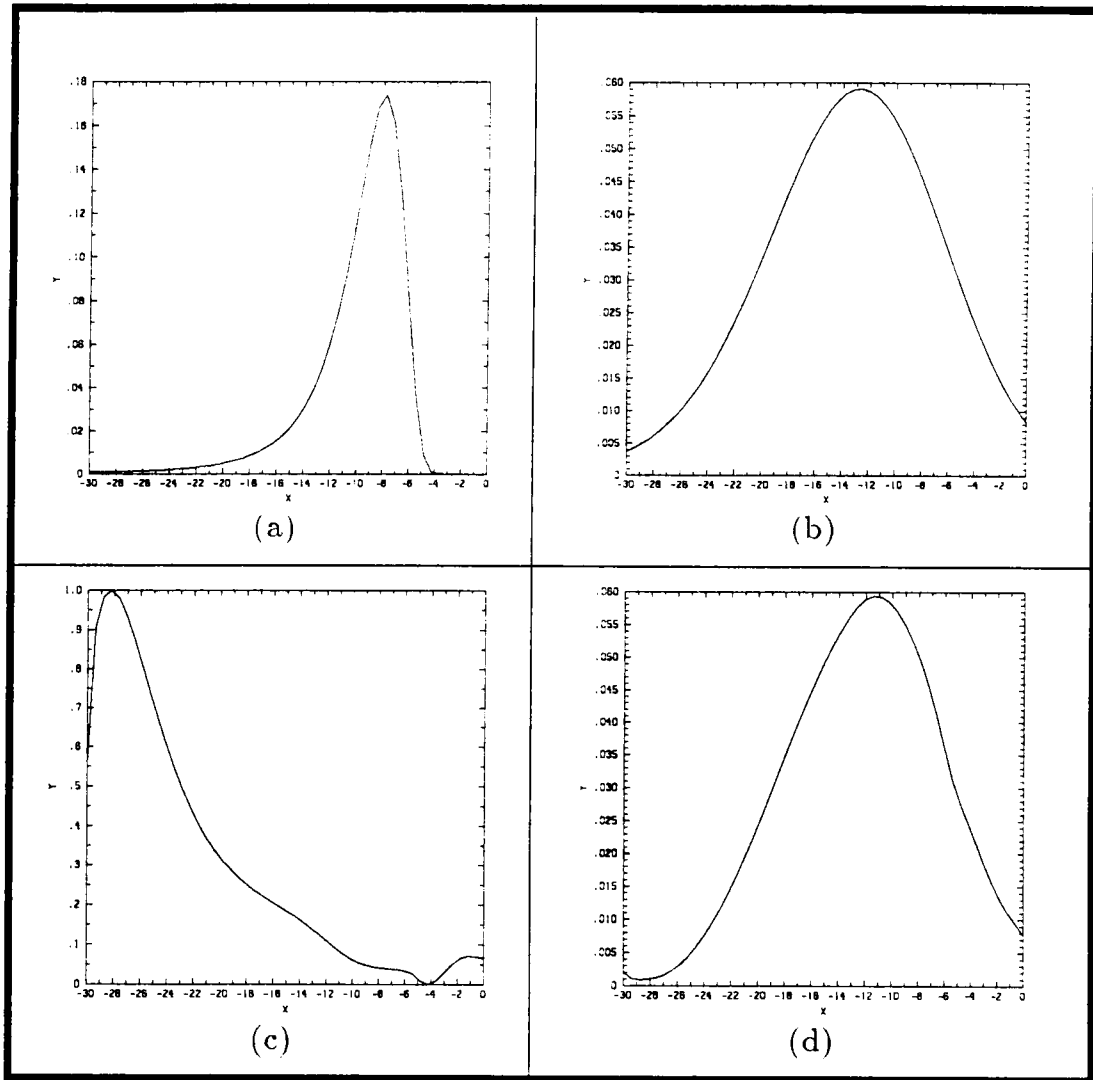


Figure 4.31: (a) p.d.f $p_{X_1}^s$ with mean $= -9.8$ and variance $= 11.8$. (b): p.d.f $p_{X_1}^r$ with mean $= -13.5$ and variance $= 38.2$. (c) Fusion function with $\lambda = 7.5$ (d) Fused density with mean -12.3 and variance 30.4

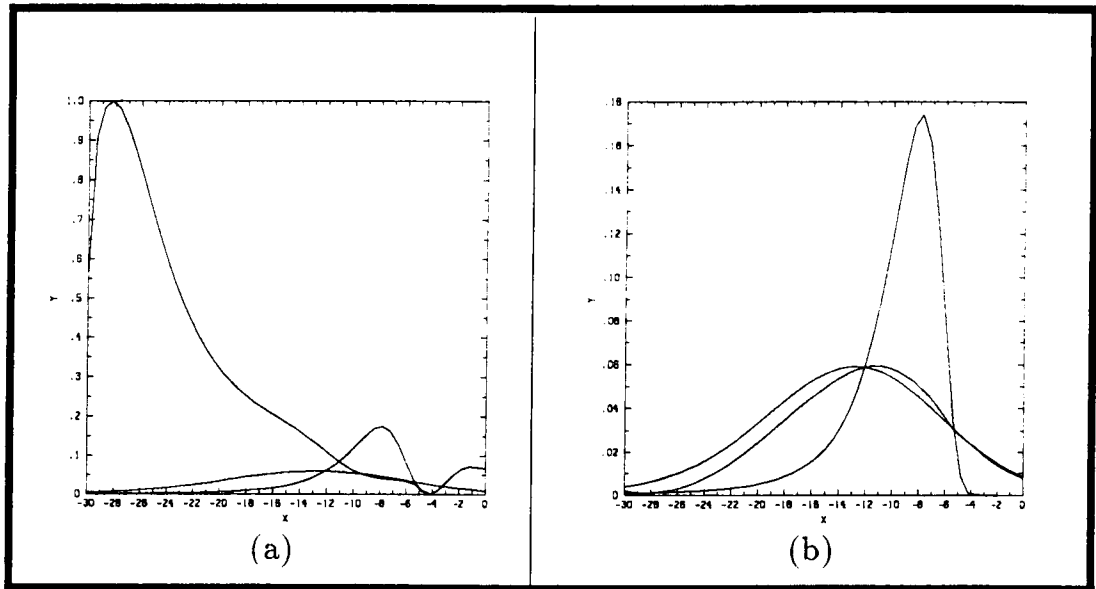


Figure 4.32: For $\lambda = 7.5$: (a) Fusion function superimposed on the two densities
(b) the fused density superimposed on the two densities

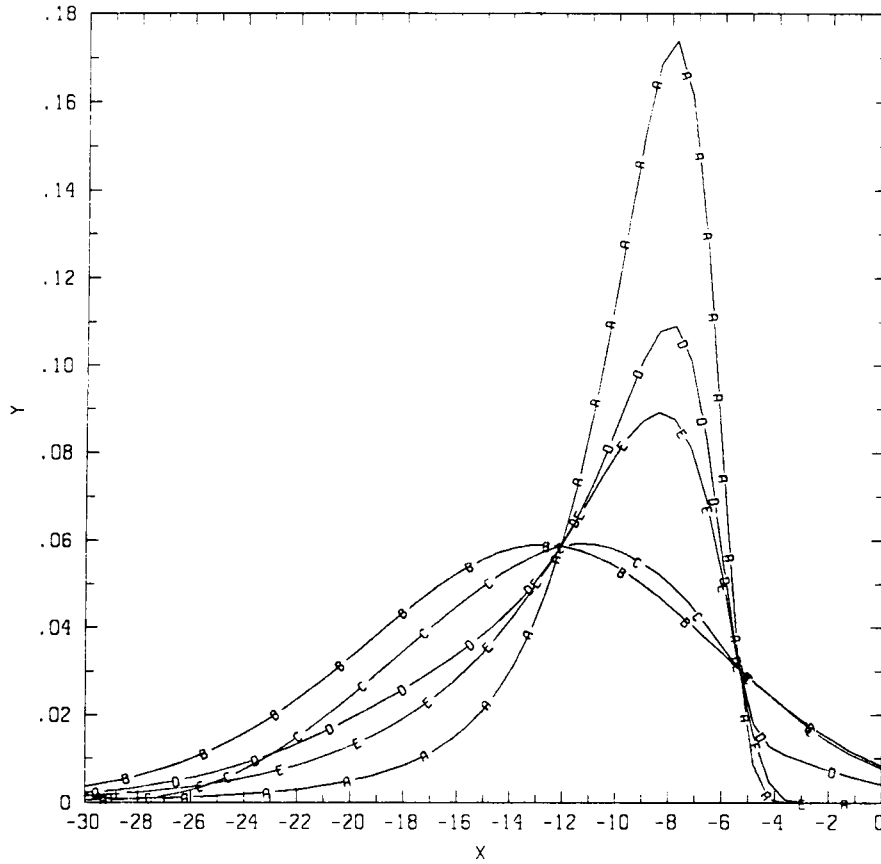


Figure 4.33: Fusion for $\lambda = 7.5$: A: Input density 1. B: Input density 2. C: Fusion of the two inputs, D: Arithmetic mean. E: Geometric mean

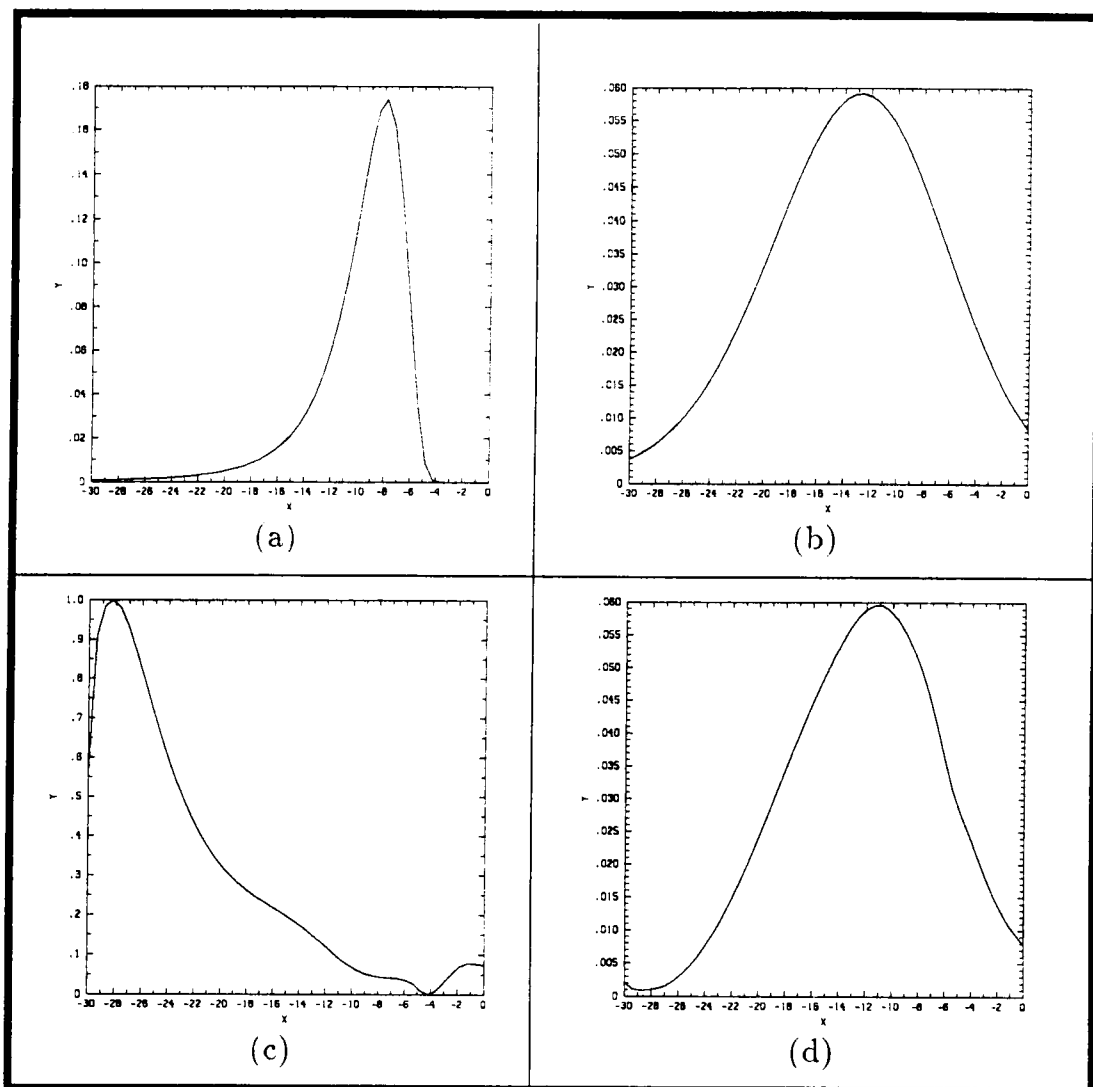


Figure 4.34: (a) p.d.f $p_{X_1}^s$ with mean $= -9.8$ and variance $= 11.8$. (b): p.d.f $p_{X_1}^r$ with mean $= -13.5$ and variance $= 38.2$. (c) Fusion function with $\lambda = 10.0$ (d) Fused density with mean -12.24 and variance 30.3

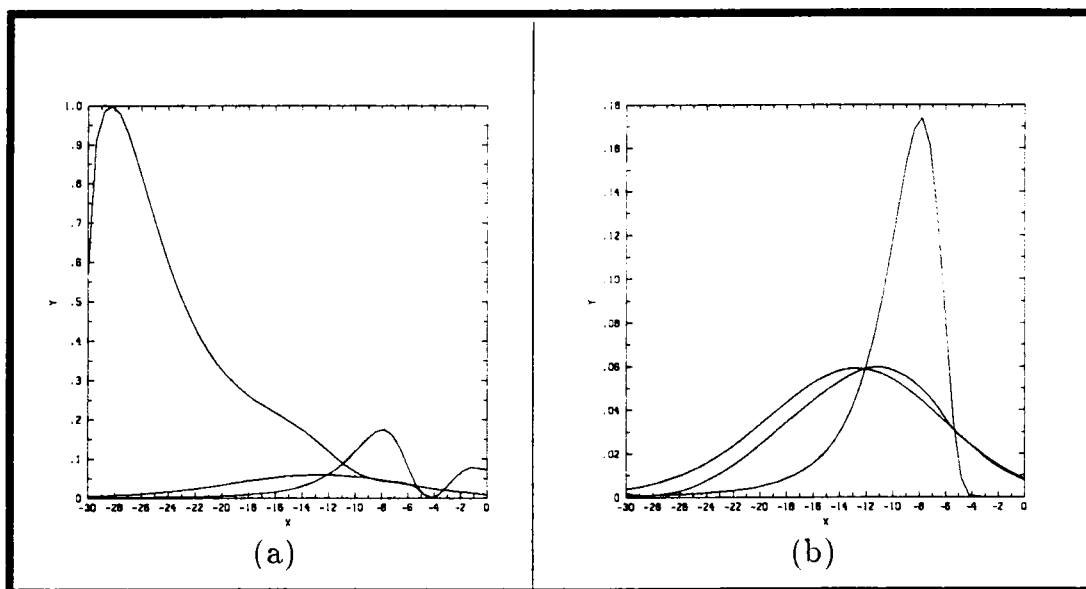


Figure 4.35: For $\lambda = 10.0$: (a) Fusion function superimposed on the two densities
 (b) the fused density superimposed on the two densities

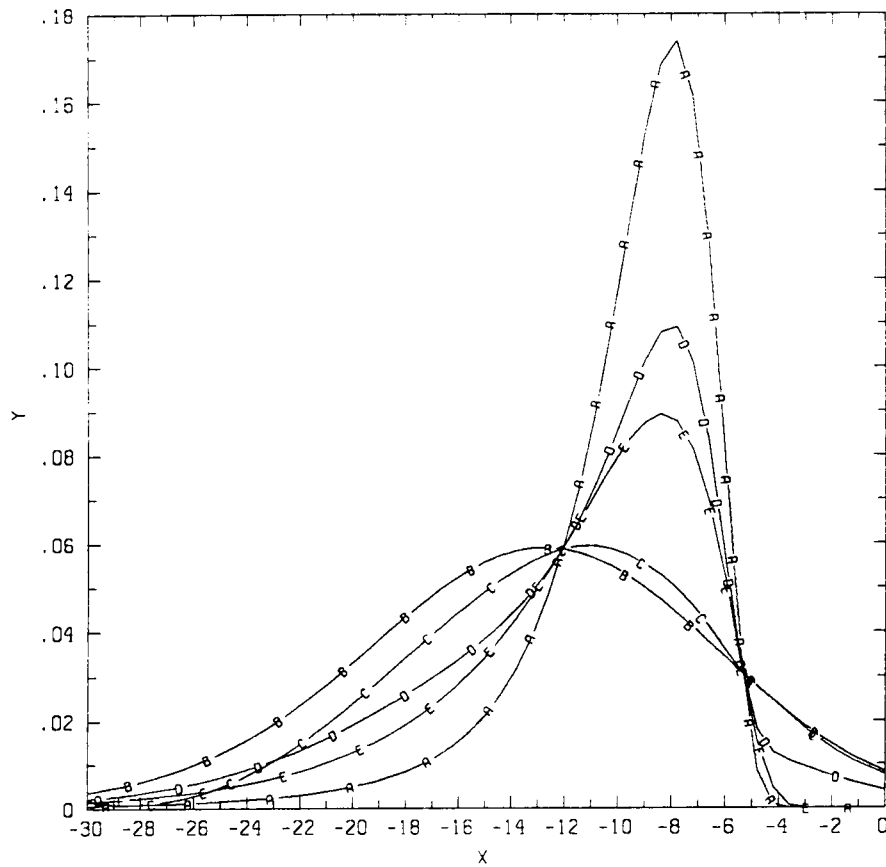


Figure 4.36: Fusion for $\lambda = 10.0$: A: Input density 1, B: Input density 2, C: Fusion of the two inputs, D: Arithmetic mean of E: Geometric mean of A

Although the principle of validity is highly stressed, the principle of corroboration is not totally eliminated. This is evident at -30 in the figures. At higher values of λ , the resulting density function behaves more than just the geometric mean of the two input densities or an "AND" operation. Figure 4.36 shows the geometric mean and the arithmetic mean of the two input densities for $\lambda = 10$.

As λ increases the variance of the fused density starts varying and the results may not be very reliable as the principle of validity is stressed more than the principle of corroboration. For $\lambda < 1$, due to the presence of oscillations the results again may not be reliable as in this case the principle of corroboration plays a major role in the fusion. So there exists an optimal λ which allows both sensor corroboration and validity. This is evident in the case for $\lambda = 1$. With reference to Fig. 4.37 to Fig. 4.39. It is seen that in the regions -30 to -20 and -4 to 0 the method allows more sensor corroboration. In the region -20 to -4 the method allows more sensor validity.

This may be illustrated as follows. For low values of λ the first order approximation of the Euler-Lagrange equation discussed in the theory may be considered to reduce to the following expression:

$$\alpha_{xx} - \left(\frac{p^s - p^r}{\lambda(p^s + p^r)} \right)^2 \alpha + \frac{2(p_x^s + p_x^r)}{p^s + p^r} \alpha_x = \frac{(1 - p^r)(p^r - p^s)}{\lambda^2(p^s + p^r)^2}.$$

From the above expression it is evident that the fusion process emphasizes $p^r - p^s$ and $1 - p^r$. For large values of λ ($\lambda > 1$) the equation reduces to the following expression:

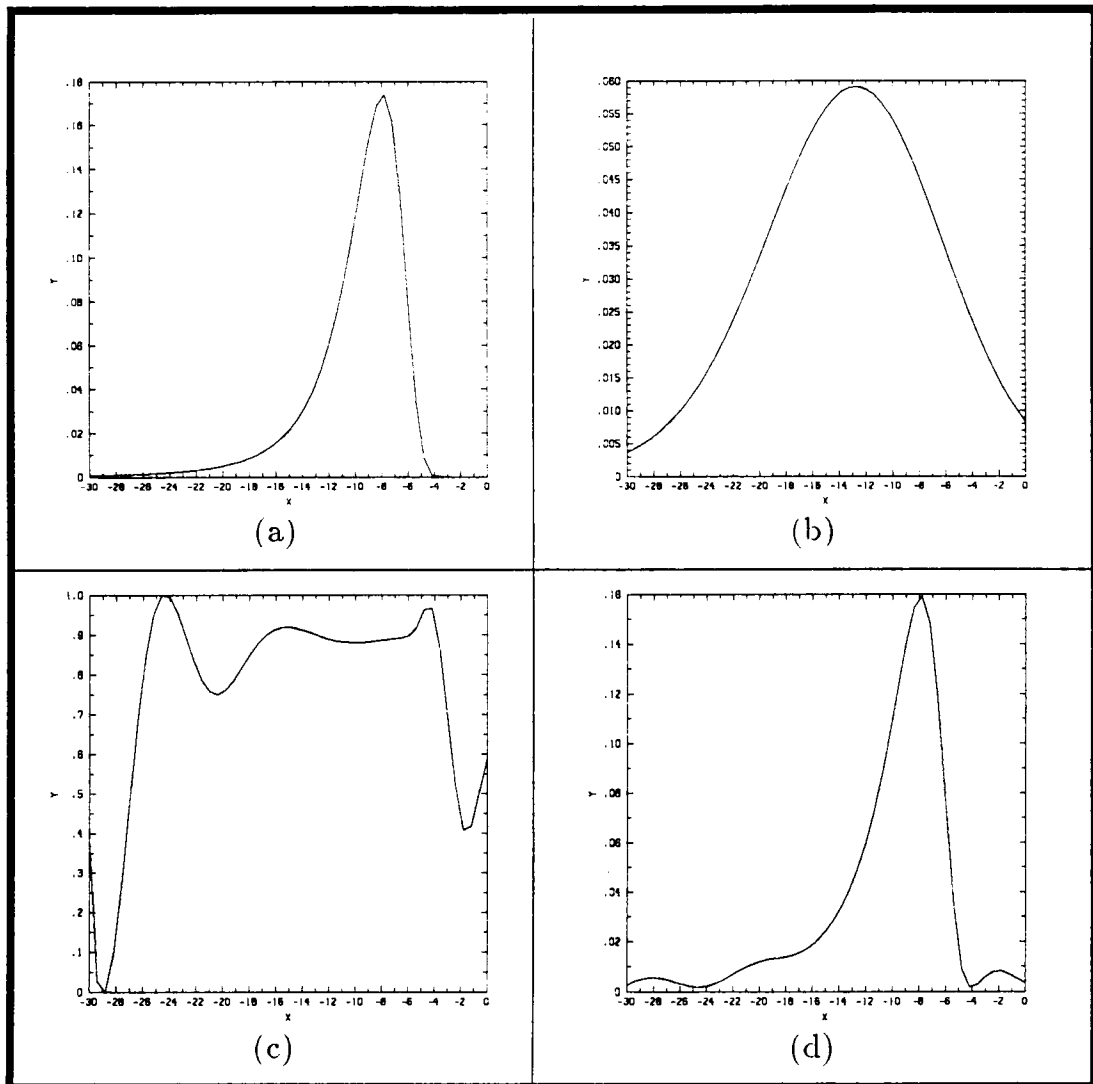


Figure 4.37: (a) p.d.f $p_{X_1}^s$ with mean $= -9.8$ and variance $= 11.8$, (b): p.d.f $p_{X_1}^r$ with mean $= -13.5$ and variance $= 38.2$. (c) Fusion function with $\lambda = 1.0$ (d) Fused density with mean -10.4 and variance 20.9

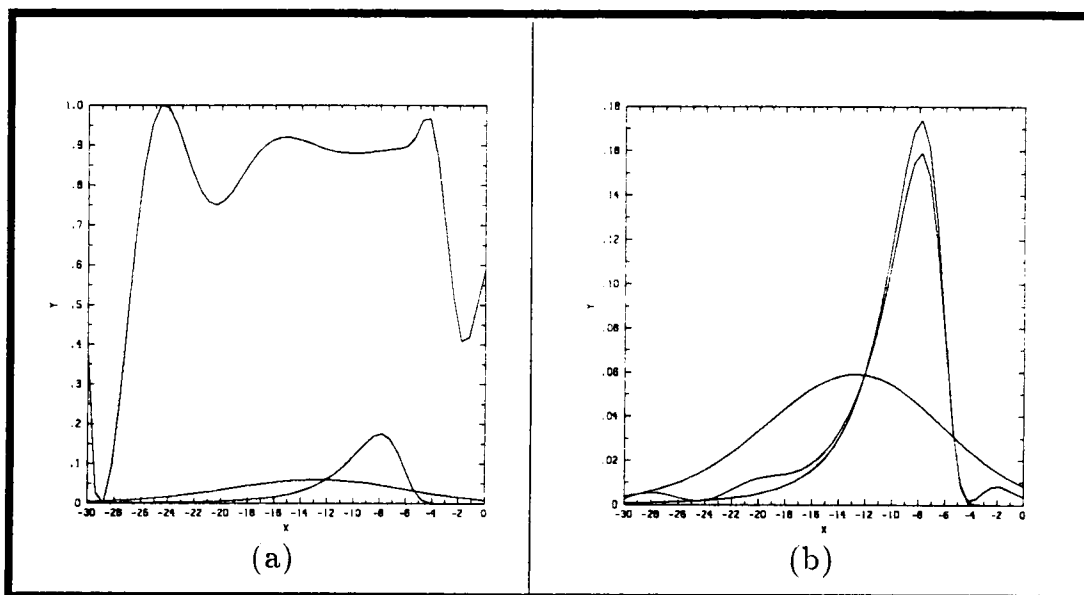


Figure 4.38: For $\lambda = 1.0$: (a) Fusion function superimposed on the two densities
 (b) the fused density superimposed on the two densities

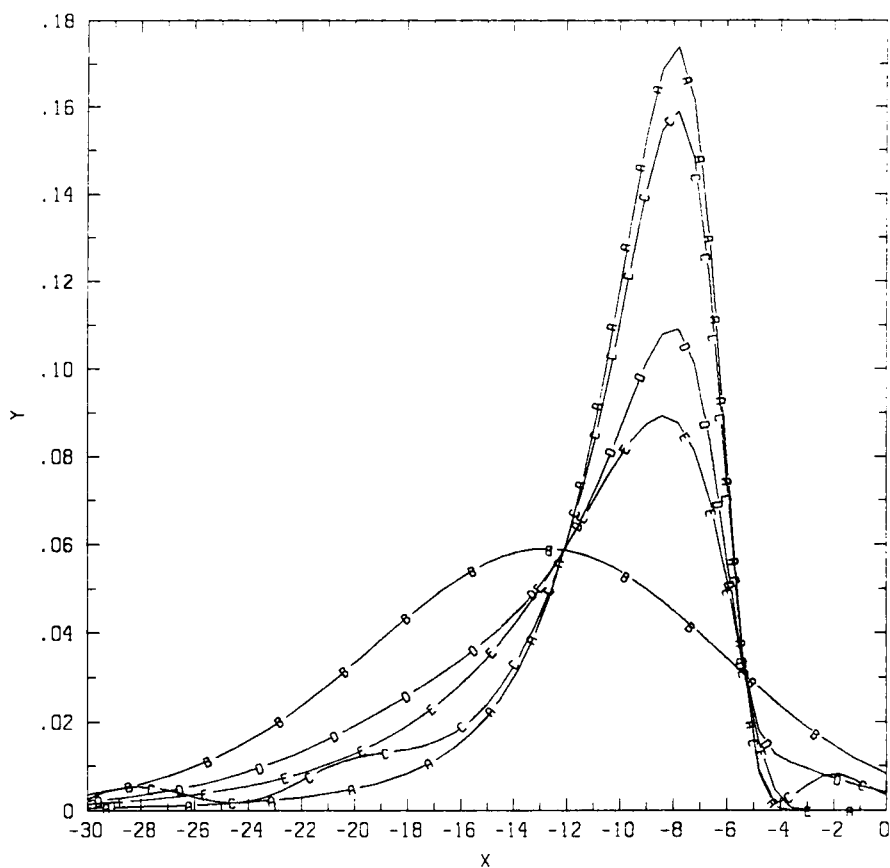


Figure 4.39: Fusion for $\lambda = 1.0$: A: Input density 1, B: Input density 2, C: Fusion of the two inputs, D: Arithmetic mean, E: Geometric mean

$$\alpha_{xx} + \frac{p_{xx}^s + p_{xx}^r}{p^s + p^r} \alpha + \frac{2(p_x^s + p_x^r)}{p^s + p^r} \alpha_x = \frac{p_{xx}^r}{p^s + p^r}.$$

Thus for large values of λ , $p^s + p^r$ is more emphasized.

The mean of the fused density function gives the X -coordinate of the 3-D point after transformation. It was found that for $\lambda = 2$ the fusion algorithm yielded both sensor corroboration and validity for the Y -coordinate. Experimentally a λ value for all the remaining coordinates of all points on the plane was chosen such that the fusion algorithm allowed both sensor corroboration and validity. The results are illustrated in Figs. 4.40 through 4.50. The means of all the densities for all the four points with the ideal positions and the associated variances are tabulated in Table 4.2 and Table 4.3, respectively.

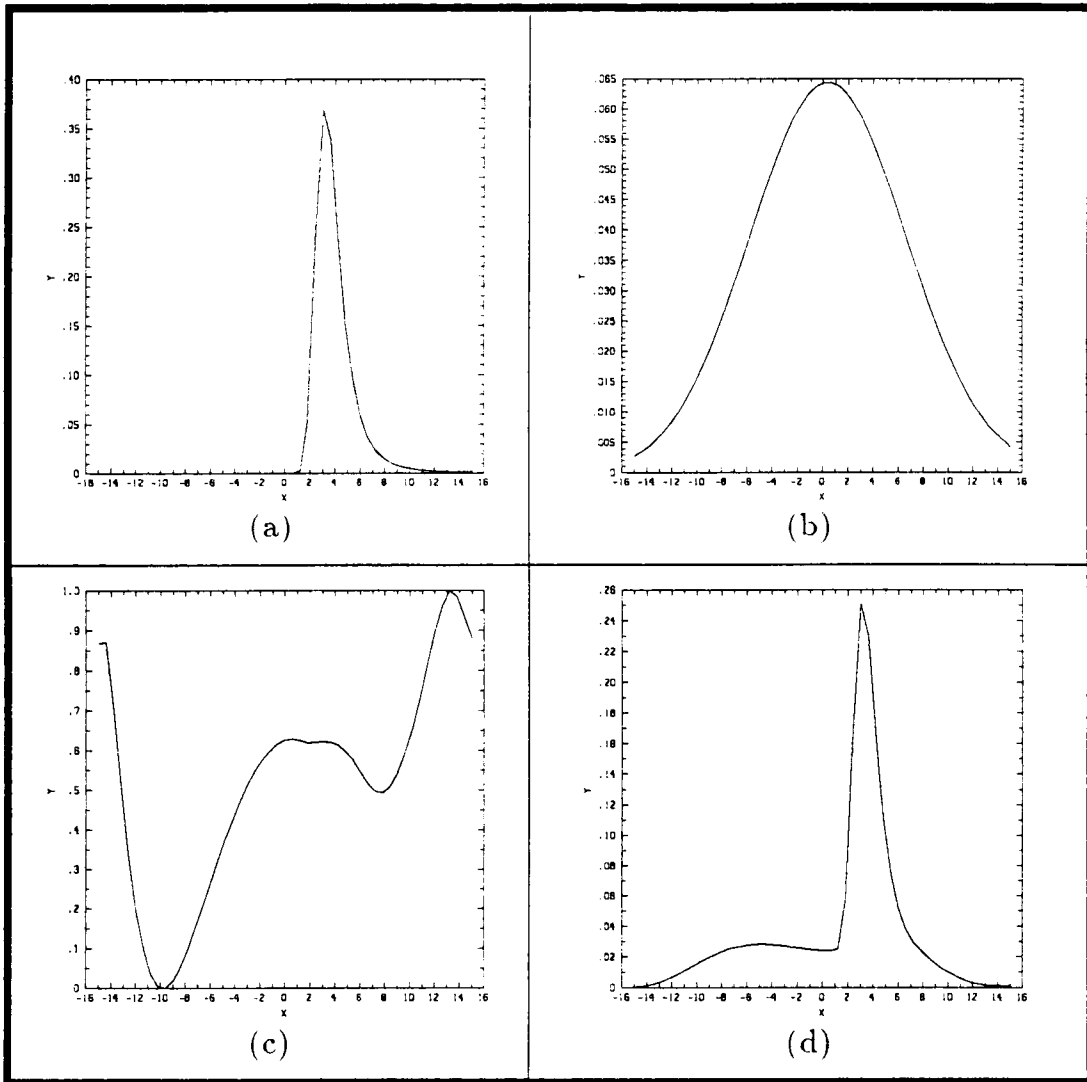


Figure 4.40: (a) p.d.f $p_{Y_1}^s$ with mean = 3.9 and variance = 2.5. (b) p.d.f $p_{Y_1}^r$ with mean = 0.3 and variance = 34.2 (c) Fusion function with $\lambda = 2.0$ (d) Fused density with mean 1.6 and variance 23.6

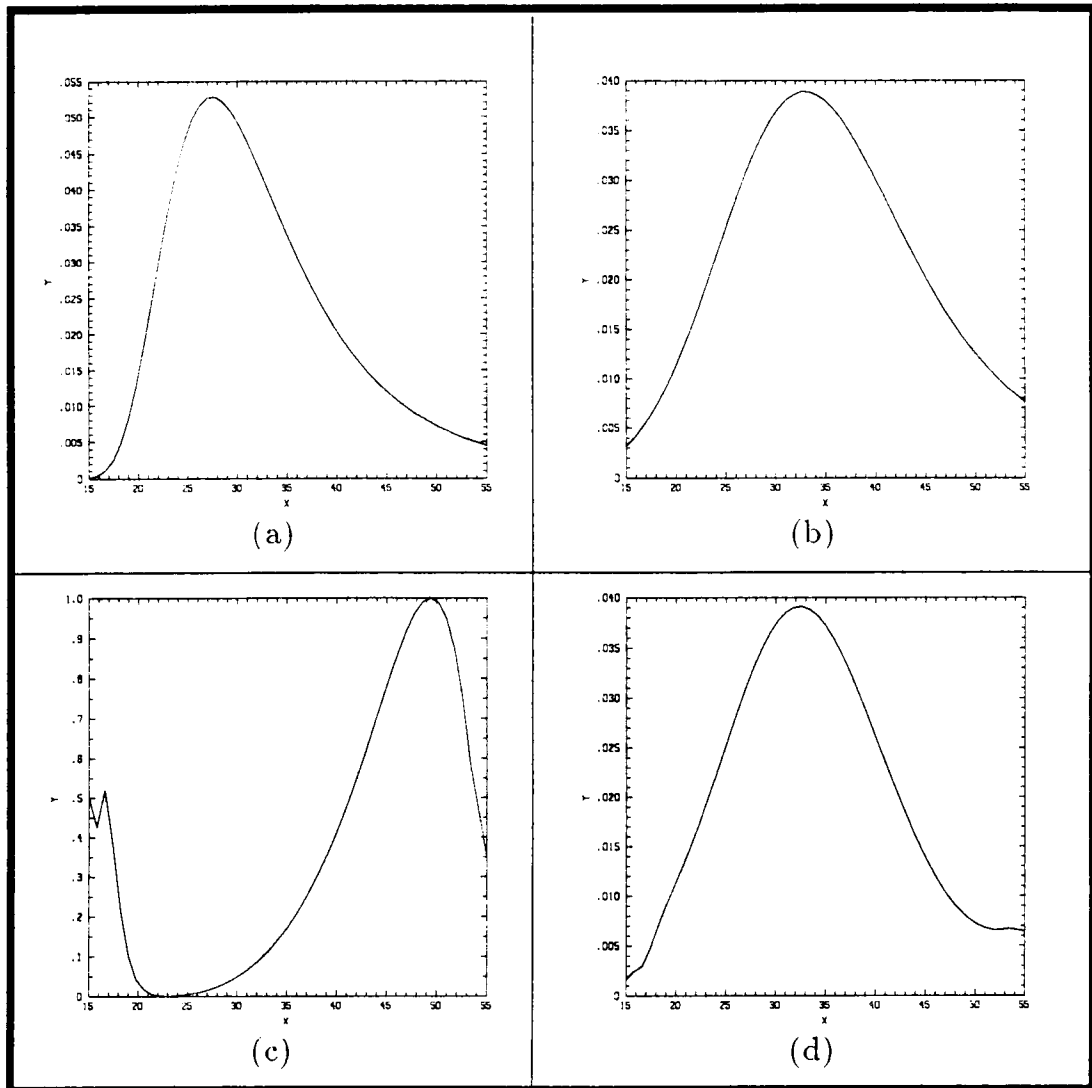


Figure 4.41: (a) p.d.f $p_{Z_1}^s$ with mean = 31.9 and variance = 61.8, (b) p.d.f $p_{Z_1}^r$ with mean = 34.8 and variance = 76.0. (c) Fusion function with $\lambda = 5.0$ (d) Fused density with mean 33.8 and variance 67.1

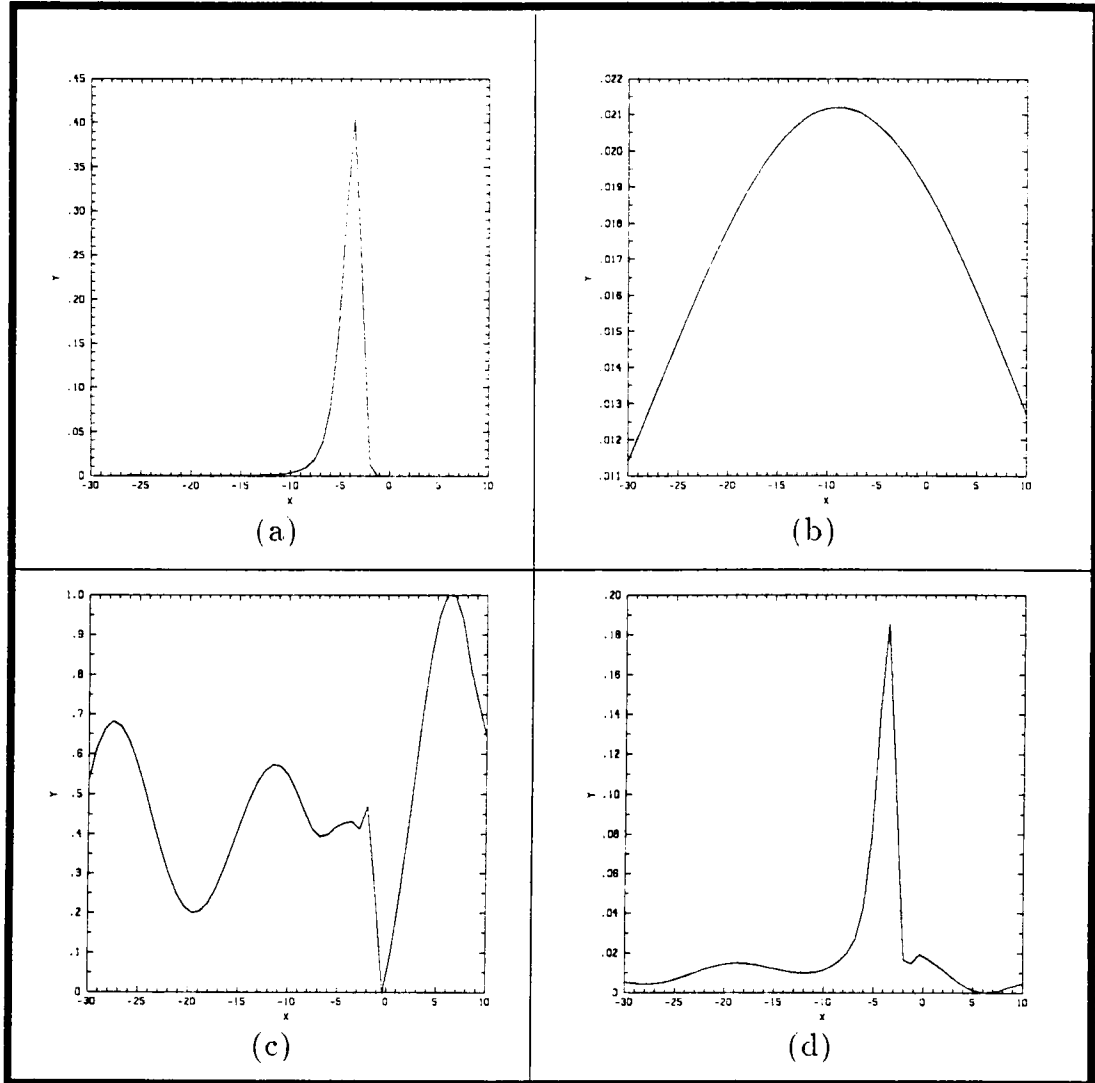


Figure 4.42: (a) p.d.f $p_{X_2}^s$ with mean = -4.3 and variance = 1.9 , (b) p.d.f $p_{X_2}^r$ with mean = -8.8 and variance = 101.8 , (c) Fusion function with $\lambda = 2.5$ (d) Fused density with mean -7.5 and variance 54.93

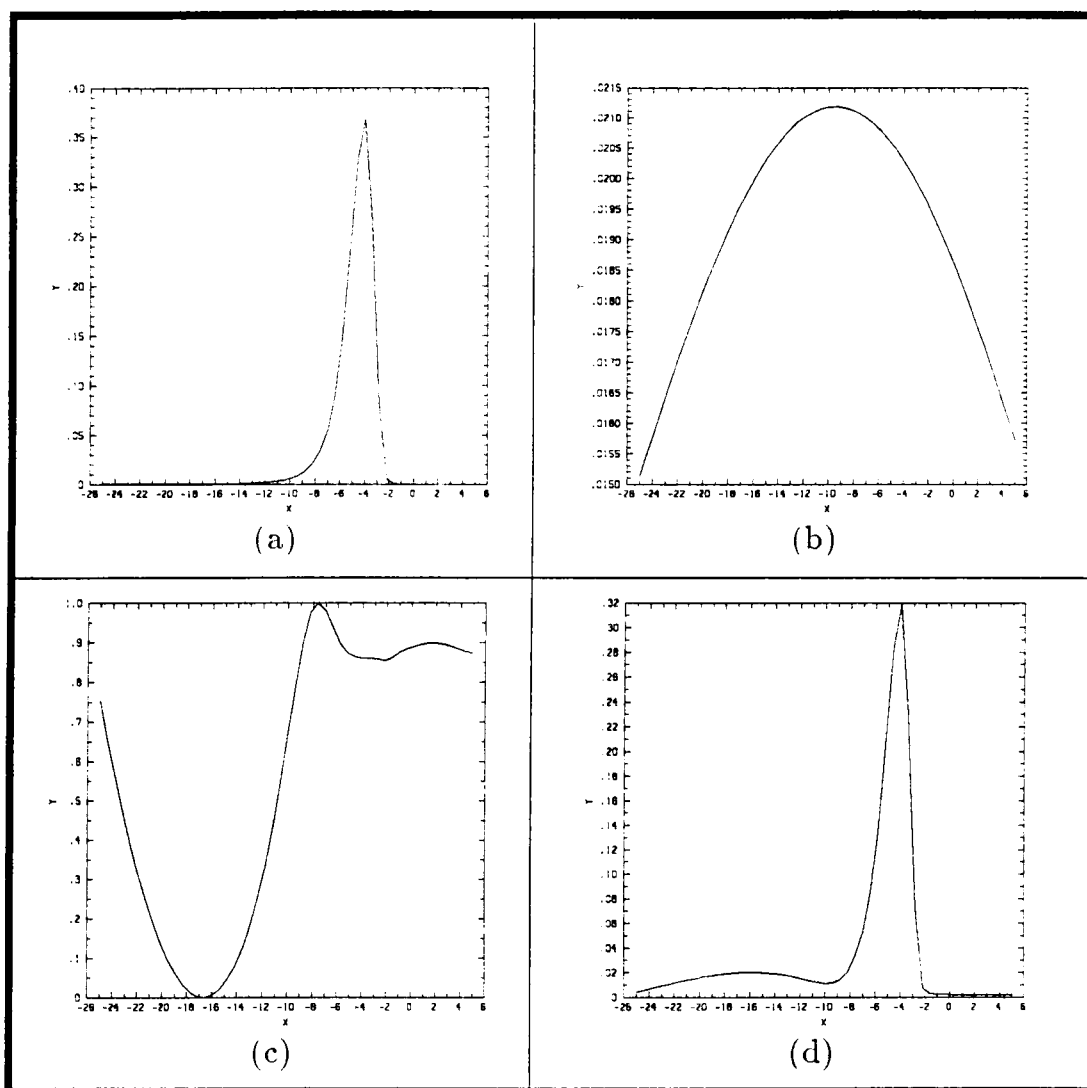


Figure 4.43: (a) $p_{Y_2}^s$, (b) $p_{Y_2}^r$, (c) Fusion function with $\lambda = 5.0$ (d) Fused density with mean -7.1 and variance 0.2

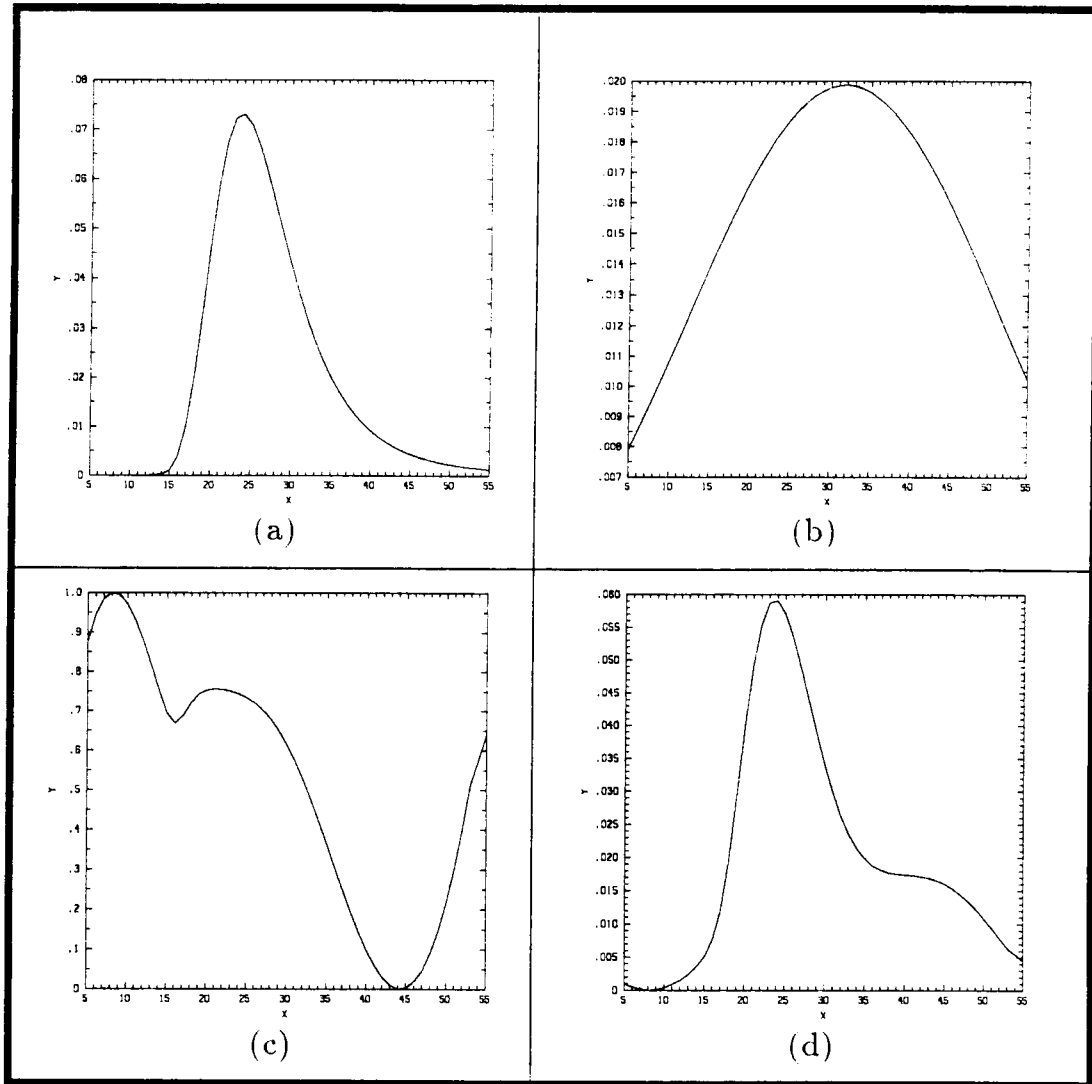


Figure 4.44: (a) p.d.f $p_{Z_2}^s$ with mean = 26.9 and variance = 44.3. (b) p.d.f $p_{Z_2}^r$ with mean = 23.8 and variance = 166.1. (c) Fusion function with $\lambda = 5.0$ (d) Fused density with mean 30.1 and variance 89.5

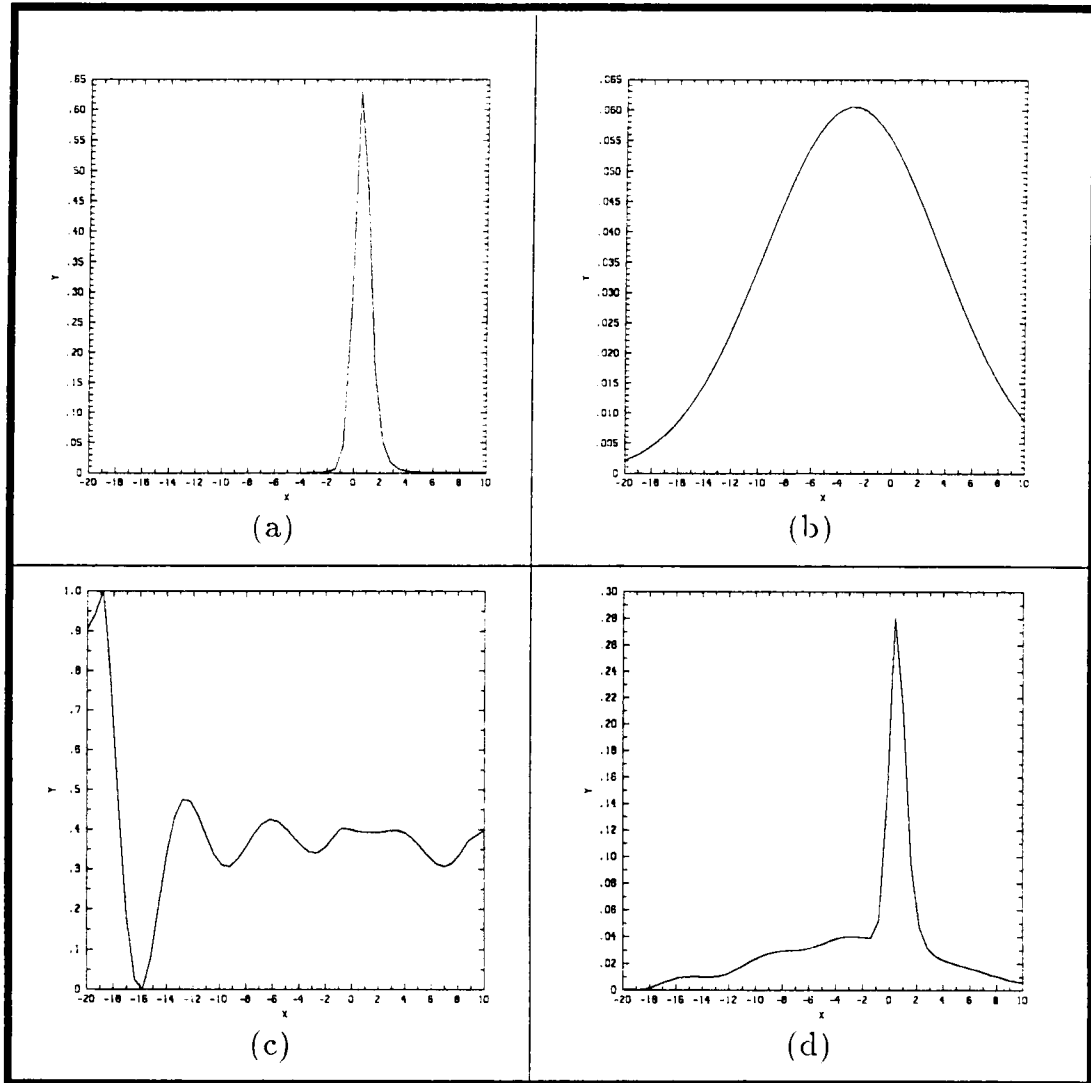


Figure 4.45: (a) p.d.f $p_{X_3}^s$ with mean = 0.7 and variance = 0.6, (b) p.d.f $p_{X_3}^r$ with mean = -3.3 and variance = 36.3, (c) Fusion function with $\lambda = 1.0$ (d) Fused density with mean -1.75 and variance 26.2

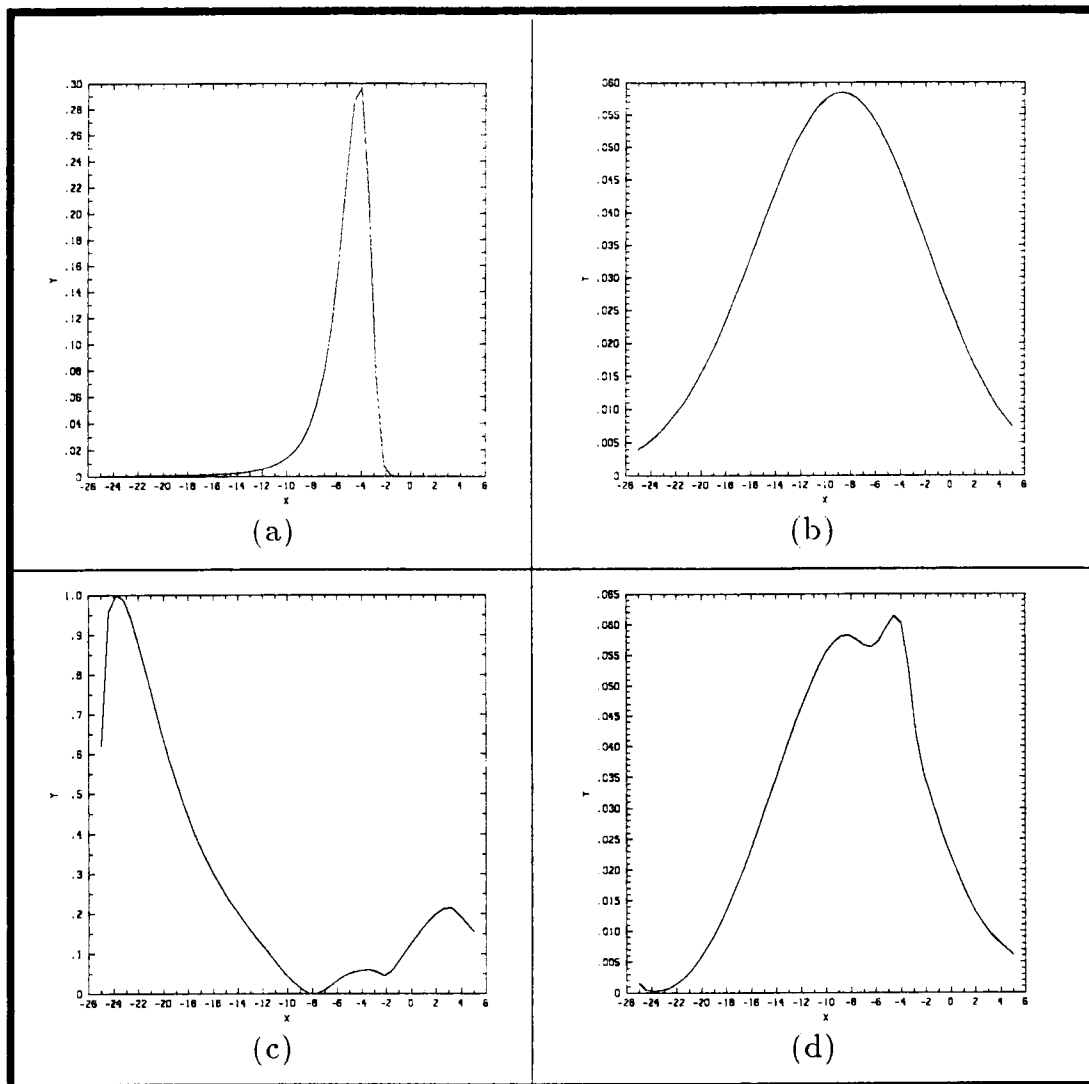


Figure 4.46: (a) $p_{Y_3}^s$, (b) $p_{Y_3}^c$. (c) Fusion function with $\lambda = 5.0$ (d) Fused density with mean -7.9 and variance 0.7

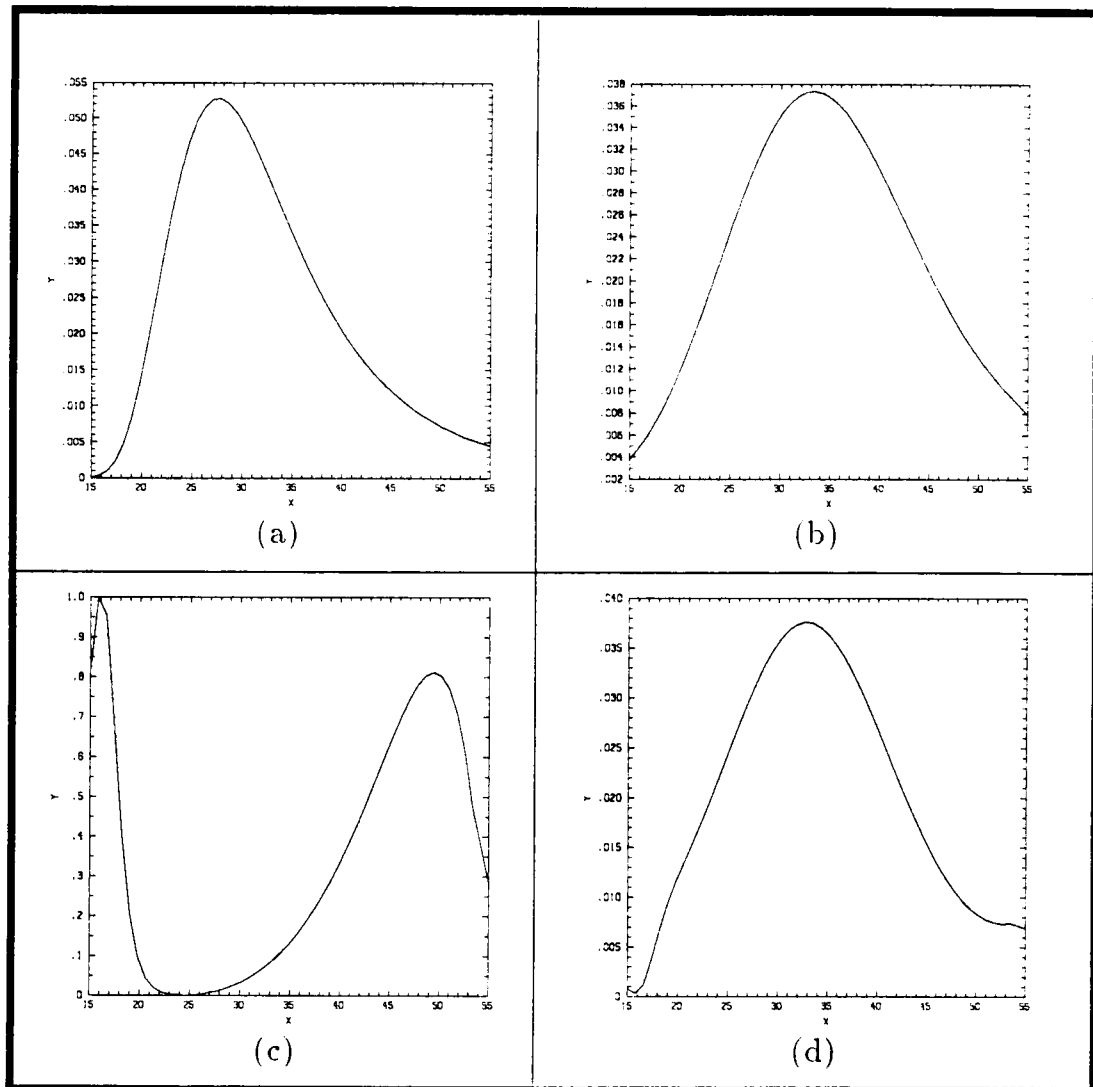


Figure 4.47: (a) p.d.f $p_{Z_3}^s$ with mean = 31.9 and variance = 61.85. (b) p.d.f $p_{Z_3}^r$ with mean = 34.8 and variance = 78.9. (c) Fusion function with $\lambda = 5.0$ (d) Fused density with mean 34.2 and variance 68.9

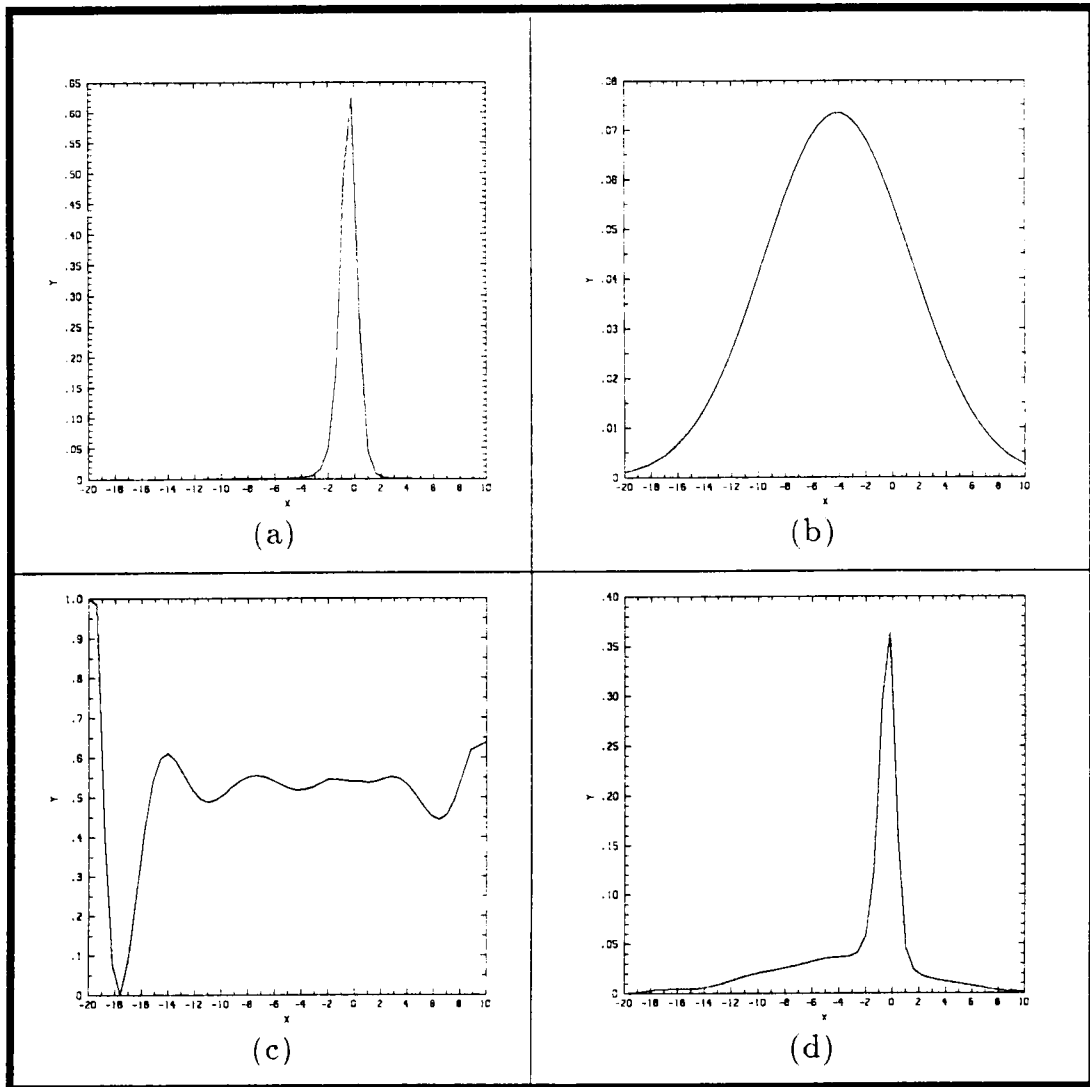


Figure 4.48: (a) p.d.f $p_{X_4}^s$ with mean = -0.46 and variance = 0.6 . (b) p.d.f $p_{X_4}^r$ with mean = -4.1 and variance = 27.8 . (c) Fusion function with $\lambda = 1.0$ (d) Fused density with mean -2.2 and variance 17.3

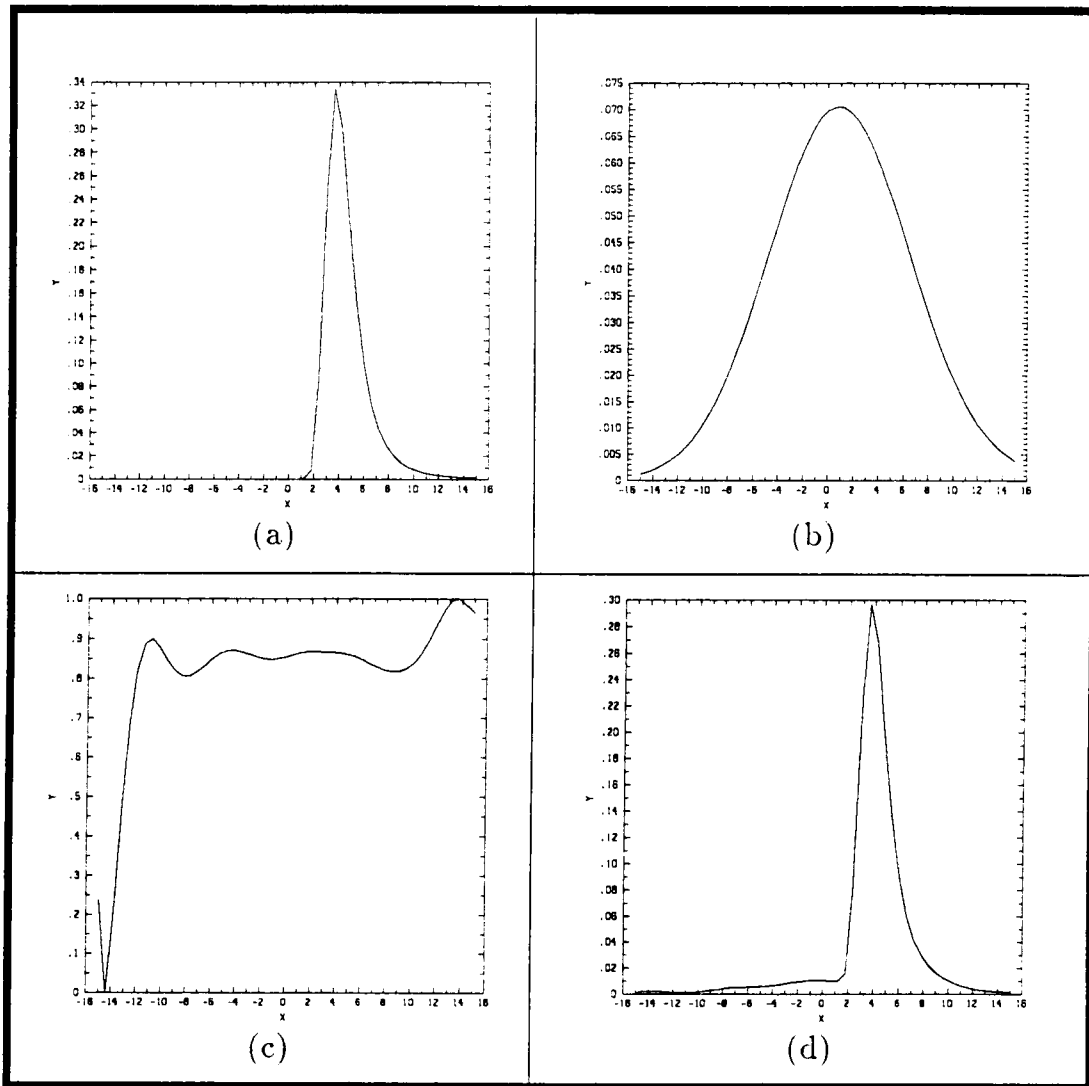


Figure 4.49: (a) p.d.f $p_{Y_4}^s$ with mean = 4.6 and variance = 2.9, (b) p.d.f $p_{Y_4}^r$ with mean = 0.8 and variance = 29.6. (c) Fusion function with $\lambda = 1.0$ (d) Fused density with mean 3.9 and variance 9.5

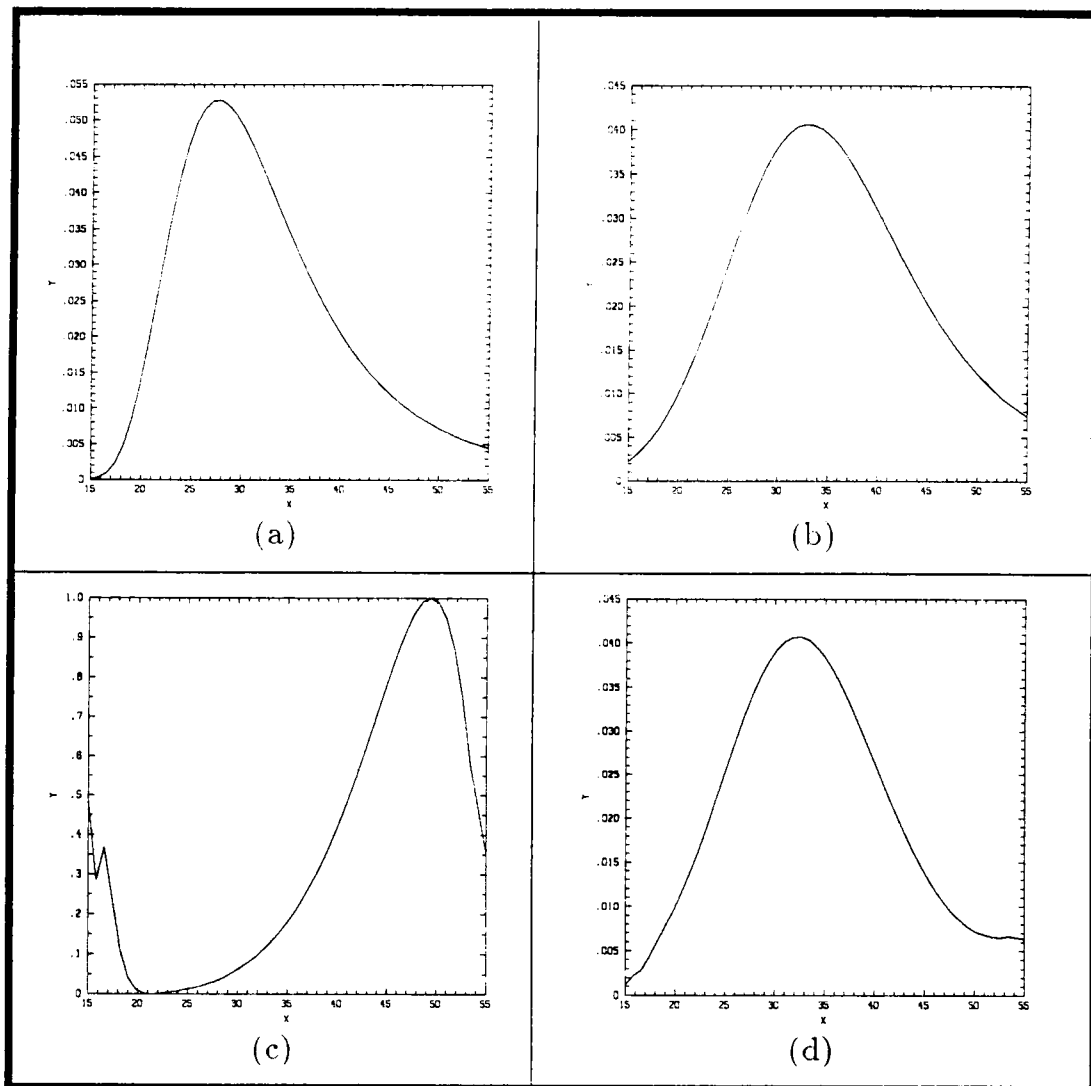


Figure 4.50: (a) p.d.f $p_{Z_4}^s$ with mean = 31.9 and variance = 61.9, (b) p.d.f $p_{Z_4}^r$ with mean = 34.9 and variance = 72.4, (c) Fusion function with $\lambda = 5.0$ (d) Fused density with mean 33.9 and variance 64.8

Table 4.2: 3 D coordinates of the plane

X				Y				Z			
Stereo	Range	Fusion	Ideal	Stereo	Range	Fusion	Ideal	Stereo	Range	Fusion	Ideal
-9.8	-13.5	-10.8	-10.2	3.9	0.35	1.6	2.0	31.9	34.8	33.7	33.9
-4.3	-8.8	-7.5	-5.6	-4.8	-9.7	-7.1	-8.0	26.9	33.2	30.1	30.2
0.67	-3.3	-1.8	-1.5	-5.3	-9.1	-7.9	-8.0	31.9	34.8	34.2	34.7
-0.5	-4.1	-2.2	-1.8	4.6	2.9	3.9	4.2	31.9	34.9	33.9	33.6

Table 4.3: Variance associated with the 3-D coordinates of the plane

σ_X			σ_Y			σ_Z		
Stereo	Range	Fusion	Stereo	Range	Fusion	Stereo	Range	Fusion
11.8	38.2	23.9	2.5	34.2	23.6	61.8	76.0	67.1
1.9	101.8	54.9	2.4	66.2	28.5	44.3	166.1	89.5
0.6	36.3	26.2	4.6	38.8	28.9	61.8	78.9	68.9
0.6	27.8	17.3	2.9	29.6	9.5	61.9	72.4	64.8

We took some uncertain sensory data from stereo and range with a variance associated at each point $(X^s, Y^s, Z^s)^T$ and $(X^r, Y^r, Z^r)^T$ as $(\sigma_x^s, \sigma_y^s, \sigma_z^s)^T$ and $(\sigma_x^r, \sigma_y^r, \sigma_z^r)^T$, respectively. Experimentally we have seen that the variances associated along the Z coordinate is very much greater than the X, Y coordinates acquired from stereo system. Conversely from range data the recovered data has larger variance in the XY component compared to Z direction. This can be attributed to the fact that in stereo, the depth of the object from the camera is inversely proportional to the disparity (distance between the two camera positions), in ultrasonic range system, it has a better resolution in measuring depth than the XY coordinates. Graphically the stereo measurement can be illustrated as an ellipsoid centered about $X^s Y^s Z^s$ with its axes as $\sigma_x^s, \sigma_y^s, \sigma_z^s$. This is illustrated in Fig. 4.51(a). Similar things can be said about the range data and is illustrated in Fig. 4.51(b). Due to similar behaviour in the XY measurement modalities Fig. 4.51 can be redrawn as Fig. 4.52. Our objective was to deal with this uncertainty in these two pieces of information by fusing the errors from stereo and range systems. For a reasonable λ the relative errors can be shown by the dotted curve in Fig. 4.52.

Traditionally the uncertainty is dealt with by simple addition of the errors associated with the measurements. This will result in the $X-Y$ and Z space much larger than the two measurements individually. This interpretation can be extended to each of the four target points from which the pose of the object can be recovered. As a result of the fusion the error on the final pose is significantly

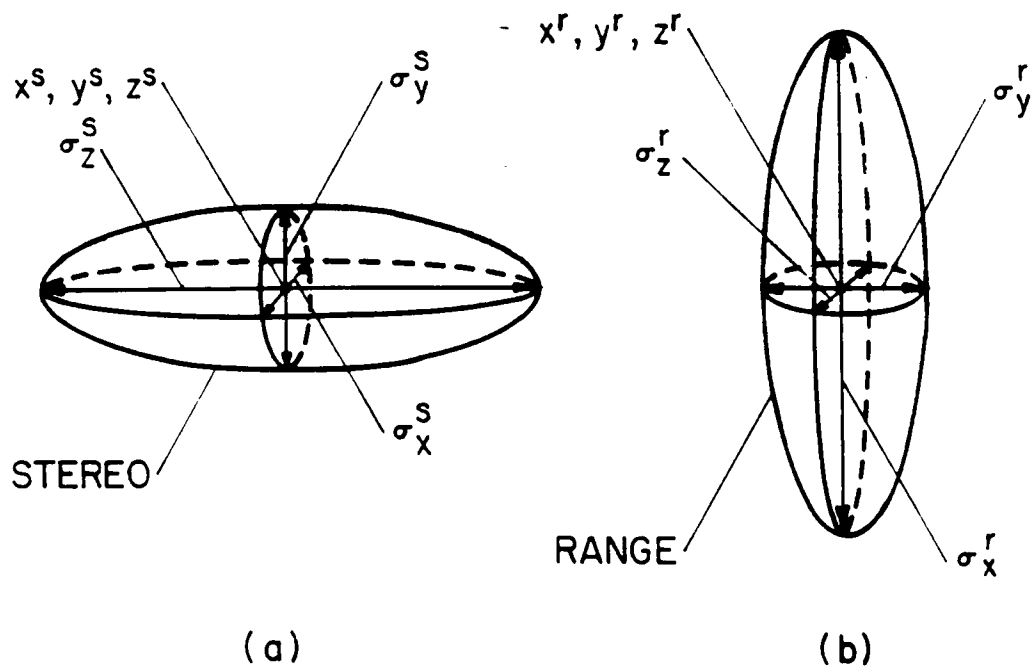


Figure 4.51: (a) Graphical representation of the variance associated to the stereo data, (b) Graphical representation of the variance associated to the range data

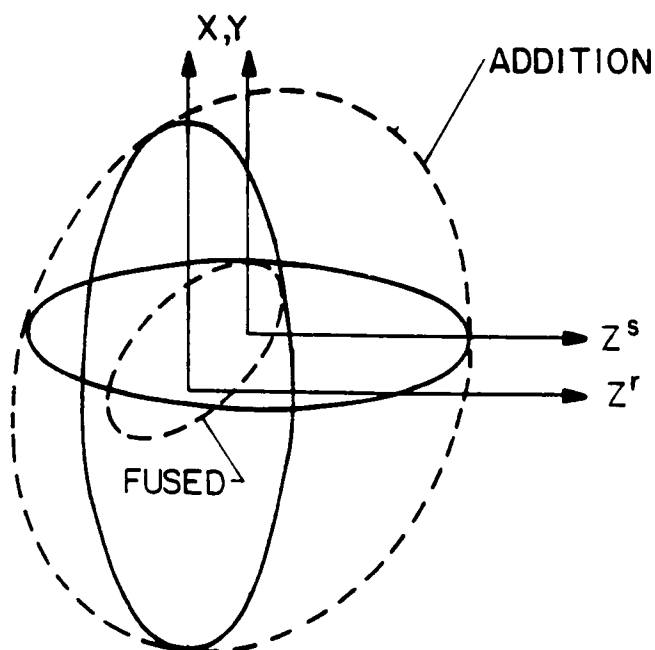


Figure 4.52: Variance in stereo, range data; the dotted curve represents the expected variance after fusing

smaller than that recovered by either sensor alone or by simply adding the errors incurred by stereo and range measurements.

CHAPTER 5

Conclusion

In this study we have described a technique to model and fuse three dimensional information from a stereo and an ultrasonic range system to establish the pose of the object and the accuracy associated with the measurements. The sources of error in a stereo system and those inherent in ultrasonic range measurement systems were probabilistically modelled.

The optimization technique used for updating the measurements generated by the two systems was solved by the Euler-Lagrange calculus of variations equations. Then the uncertainties inherent in each of the measurements were combined by fusing the probability densities of the stereo and ultrasonic range for each of the target points describing the pose.

This method in dealing with uncertainty in data extracted from disparate sensors is very advantageous in that it exploits the best features and ignores the weaknesses of each sensor. Thus it has enabled us to obtain unique information (3-D pose of an object) for significantly different data.

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] J. M. Tenenbaum, "On Locating Objects by Their Distinguishing Features in Multisensory Images." *Computer Graphics and Image Processing*, vol. 2, no. 3, pp. 308-320, December 1973.
- [2] T. D. Garvey, "An Experiment With A System for Locating Objects in Multisensory Images." *Proceedings of Third International Joint Conference on Pattern Recognition*, pp. 567-575, November 1976.
- [3] M. J. Magee and J. K. Aggarwal, "Using Multisensory Images to Derive the Structure of Three-Dimensional Objects - A Review." *Computer Vision, Graphics, and Image Processing*, vol. 32, pp. 145-157, 1985.
- [4] J. H. Graham, "An Evidential Approach to Robot Sensory Fusion." *Proceedings of the 1986 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 492-497, October 14-17 1986.
- [5] O. D. Faugeras and M. Hebert, "The Representation, Recognition, and Locating of 3-D Objects," *The International Journal of Robotics Research*, vol. 5, no. 3, pp. 27-51, Fall 1986.
- [6] Y. Hung, P. Yeh, and D. Harwood, "Passive Ranging To Known Planar Point Sets," *IEEE International Conference on Robotics and Automation*, pp. 80-85, March 1985.
- [7] Y. Yakimovsky and R. Cunningham, "A System for Extracting Three-Dimensional Measurements from a Stereo Pair of TV Cameras," *Computer Graphics and Image Processing*, vol. 2, pp. 195-209, April 1978.
- [8] M. D. Levine, D. A. O'Handley, and G. M. Yagi, "Computer Determination of Depth Maps." *Computer Graphics and Image Processing*, vol. 2, no. 2, pp. 131-150, October 1973.
- [9] R. Nevatia, "Depth Measurement by Motion Stereo," *Computer Graphics and Image Processing*, vol. 5, no. 2, pp. 203-214, June 1976.
- [10] R. O. Duda, D. Nitzan, and P. Barrett, "Use of Range and Reflectance Data to Find Planar Surface Regions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 3, pp. 259-271, July 1979.
- [11] R. O. Duda and D. Nitzan, "Low-Level Processing of Registered Intensity and Range Data," *Proc. of Third International Joint Conference on Pattern Recognition*, pp. 598-601, November 1976.

- [12] W. E. L. Grimson. "Computational Experiments with a Feature Based Stereo Algorithm." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-7, no. 1, pp. 17-33, January 1985.
- [13] R. C. Gonzalez and P. Wintz. *Digital Image Processing*. Reading, MA: Addison-Wesley Publishing Company, 1987.
- [14] K. S. Fu, R. C. Gonzalez, and C. S. G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*. New York, NY: McGraw - Hill Book Company, 1987.
- [15] B. K. P. Horn. *Robot Vision*. Cambridge, MA: The MIT Press, 1986.
- [16] R. O. Duda and P. E. Hart, *Pattern Classification And Scene Analysis*. New York, NY: John Wiley and Sons, 1973.
- [17] V. Torre, A. Verri, and A. Fiumicelli. "Stereo Accuracy for Robotics," in *Third International Symposium on Robotics Research*, (O. D. Faugeras and G. Giralt, eds.), Cambridge, Massachusetts: The MIT Press, 1986.
- [18] Z. Huang, "Error Analysis of Position and Orientation in Robot Manipulators," *Mechanical Machine Theory*, vol. 22, no. 6, pp. 577-581, 1987.
- [19] E. S. McVey and J. W. Lee, "Some Accuracy and Resolution Aspects of Computer Vision Distance Measurements," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-4, pp. 646-649, November 1982.
- [20] S. D. Blostein and T. S. Huang, "Error Analysis in Stereo Determination of 3-D Point Positions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 6, pp. 752-765, November 1987.
- [21] R. A. Jarvis, "A Perspective on Range Finding Techniques for Computer Vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-5, no. 2, pp. 122-139, March 1983.
- [22] B. A. Boyter, "Three-Dimensional Matching Using Range Data," *IEEE Computer Society: First Conference on Artificial Intelligence Applications*, pp. 211-216, December 1984.
- [23] B. C. Vemuri and J. K. Aggarwal. "Resolving the Orientation and Identity of an Object from Range Data." *Spatial Reasoning and Multi-Sensor Fusion: Proceedings of the 1987 Workshop*, pp. 178-187, October 1987.
- [24] M. J. Magee, B. A. Boyter, C. Chien, and J. K. Aggarwal, "Experiments in Intensity Guided Range Sensing Recognition of Three-Dimensional Objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-7, no. 6, pp. 629-637, November 1985.

- [25] J. K. Aggarwal and M. J. Magee. "Determining Motion Parameters Using Intensity Guided Range Sensing." *Pattern Recognition*, vol. 19, no. 2, pp. 169-180, April 1986.
- [26] B. Gil, A. Mitiche, and J. K. Aggarwal. "Experiments in Combining Intensity and Range Edge Maps." *Computer Vision, Graphics, and Image Processing*, vol. 21, no. 3, pp. 395-411, March 1983.
- [27] S. Y. Harmon, G. L. Bianchini, and B. E. Pinz. "Sensor Data Fusion Through A Distributed Blackboard." Tech. Rep. code 442, Naval Ocean System Center, San Diego, CA, 1982.
- [28] E. W. Kent, O. Shneier, and H. Tsai-Hong, "Building Representation From Fusions of Multiple Views," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1634-1639, April 1986.
- [29] C. Delcroix, *Analytical Fusion of Edge Maps in Multisensory Images*. Master's thesis, The University Of Tennessee, Knoxville, August 1988.
- [30] M. K. Brown, "Feature Extraction Techniques for Recognizing Solid Objects with an Ultrasonic Range Sensor," *IEEE Journal of Robotics and Automation*, vol. RA-1, no. 4, pp. 191-205, December 1985.
- [31] G. D. Maslin, "A Simple Ultrasonic Ranging System," *102nd Convention of the Audio Engineering Society*, May 12 1983.
- [32] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. New York, NY: McGraw - Hill Book Company, 1984.
- [33] M. A. Abidi and C. J. Delcroix, "Analytic Fusion of Edge Maps," *Proceedings IEEE Southeastcon*, vol. 2, pp. 739-744, April 9-12 1989.
- [34] R. Weinstock, *Calculus of Variations*. New York, NY: Dover Publications Inc., 1974.
- [35] D. A. Anderson, J. C. Tannehill, and R. H. Pletcher, *Computational Fluid Mechanics and Heat Transfer*. New York, NY: McGraw - Hill Book Company, 1984.

Vita

Ravindra Vaman Narkhede was born in Karwar, India, on April 4, 1960. He graduated with a Bachelor of Engineering degree with major in Electrical Engineering with specialization in Control Systems and Electrical Machinery in December, 1983.

During this course of study, he took part in the Indo-British Universities Research Cooperation Program as a research assistant at the Indian Institute of Sciences, Bangalore, India. He completed his bachelors thesis titled "Automatic Speed Control of a D.C Motor". He worked for two years in the industry upon graduation as a Maintenance Engineer. For a while he also worked in the Artificial Intelligence laboratory of the University of Kaiserslautern, West Germany.

He joined the Electrical and Computer Engineering Department at the University of Tennessee as a graduate student in September of 1985. He was a research assistant with the Artificial Intelligence group at the Oak Ridge National Laboratory from November 1985 to May 1987. He then took on a position as a Teaching assistant with the Nuclear Engineering department to pursue his own research in Robotics under the supervision of Dr. M. A. Abidi. The author is a member of IEEE and Eta Kappa Nu Honor Society.