A COMPARISON OF DIFFERENT APPROACHES TO TARGET DIFFERENTIATION WITH SONAR

A DISSERTATION

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCES

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

By

Birsel Ayrulu (Erdem)

June 2001

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

Billur Barshan, Ph. D. (Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

Ömer Morgül, Ph. D.

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

A. Enis Çetin, Ph. D.

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

Ülkü Gürler, Ph. D.

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

Yasemin Yardımcı Çetin, Ph. D.

Approved for the Institute of Engineering and Sciences:

Prof. Dr. Mehmet Baray
Director of Institute of Engineering and Sciences

A COMPARISON OF DIFFERENT APPROACHES TO TARGET DIFFERENTIATION WITH SONAR

Birsel Ayrulu (Erdem)
Ph.D. in Department of Electrical and Electronics Engineering
Supervisor: Assoc. Prof. Dr. Billur Barshan
June 2001

This study compares the performances of different classification schemes and fusion techniques for target differentiation and localization of commonly encountered features in indoor robot environments using sonar sensing. Differentiation of such features is of interest for intelligent systems in a variety of applications such as system control based on acoustic signal detection and identification, map-building, navigation, obstacle avoidance, and target tracking. The classification schemes employed include the target differentiation algorithm developed by Ayrulu and Barshan, statistical pattern recognition techniques, fuzzy c-means clustering algorithm, and artificial neural networks. The fusion techniques used are Dempster-Shafer evidential reasoning and different voting schemes. To solve the consistency problem arising in simple majority voting, different voting schemes including preference ordering and reliability measures are proposed and verified experimentally. To improve the performance of neural network classifiers, different input signal representations, two different training algorithms, and both modular and non-modular network structures are considered. The best classification and localization scheme is found to be the neural network classifier trained with the wavelet transform of the sonar signals. This method is applied to map-building in mobile robot environments. Physically different sensors such as infrared sensors and structured-light systems besides sonar sensors are also considered to improve the performance in target classification and localization.

Keywords: Sonar sensing, target differentiation, target localization, artificial neural networks, learning, feature extraction, statistical pattern recognition, Dempster-Shafer evidential reasoning, majority voting, sensing systems, acoustic signal processing, mobile robots, map-building, Voronoi diagram.

ÖZET

SONARLA HEDEF AYIRDETMEDE FARKLI YÖNTEMLERİN KARŞILAŞTIRILMASI

Birsel Ayrulu (Erdem)
Elektrik ve Elektronik Mühendisliği Doktora
Tez Yöneticisi: Doç. Dr. Billur Barshan
Haziran 2001

Bu çalışmada, akustik algılayıcılar kullanılarak farklı sınıflandırma ve tümleştirme yöntemlerinin gezer robot ortamlarında sıklıkla karşılaşılan hedefleri birbirinden ayırdetme ve konumlarını kestirmedeki başarımları karşılaştırılmıştır. Bu tür hedeflerin ayırdedilmesi akustik sinyallerin sezimi ve tanınmasına dayalı sistem denetimi, harita çıkarımı, engel bertarafı, hedef izleme gibi uygulamalarla yakından ilgilidir. Kullanılan sınıflama yöntemleri Ayrulu ve Barshan tarafından geliştirilen bir sınıflandırma algoritması, istatistiksel örüntü tanıma yöntemleri, bulanık c-ortalama kümelendirme algoritması ve yapay sinir ağlarını içermektedir. Kullanılan tümleştirme yöntemleri, Dempster-Shafer kanıtsal akıl yürütme ve çeşitli oylama yöntemleridir. Basit çoğunluk oylamasında görülen tutarlılık sorununun çözümü için tercih sıralaması ve değişik güvenilirlik ölçütlerini içeren farklı oylama yöntemleri önerilmiş ve deneysel olarak uygulanabilirliği gösterilmiştir. Yapay sinir ağlarının başarımını arttırmak amacıyla farklı girdi sinyal dönüşümleri, iki ayrı eğitme algoritması ve modüler ve modüler olmayan ağ yapıları denenmiştir. Elde edilen en iyi sınıflandırma yöntemi olan ve sonar sinyallerinin dalgacık dönüşümünü kullanan yapay sinir ağı, gezer robot ortamlarının harita çıkarımında deneysel olarak kullanılmıştır. Hedef sınıflandırma ve konum kestirimindeki başarımın artırılması amacıyla, akustik algılayıcılara ek olarak kızıl ötesi algılayıcılar ve yapılandırılmış-ışık sistemleri gibi farklı fiziksel yaıpıdaki algılayıcı sistemleri de kullanıldı.

Anahtar Kelimeler: Akustik algılama, hedef sınıflandırma ve konum kestirimi, yapay sinir ağları, öğrenme, özellik çıkarımı, istatistiksel örüntü tanıma, Dempster-Shafer kanıtsal akıl yürütme, çoğunluk oylaması, algılayıcı sistemleri, akustik sinyal işleme, gezer robotlar, harita çıkarımı, Voronoi diyagram.

ACKNOWLEDGMENTS

I would like to thank everyone who contributed to this thesis. First of all, I would like to express my sincere thanks to my thesis supervisor Dr. Billur Barshan for her supervision, guidance, suggestions and encouragement throughout the development of this thesis.

I would also like to thank Dr. Ömer Morgül, Dr. Levent Onural, and Dr. Ülkü Gürler for reading, commenting, and making useful suggestions on my thesis proposal report in the early stages of this thesis.

I would also like to thank the members of my committee: Dr. Ömer Morgül, Dr. Enis Çetin, Dr. Ülkü Gürler, and Dr. Yasemin Yardımcı Çetin for reading and commenting on my thesis.

I would like to express my sincere thanks to Lütfiye Durak for her cordial friendship. She has done many things for me. It is not possible to state all of them here, but I know that I have indebted many things to her especially in my last two years at Bilkent.

I also thank Çağatay Candan for providing the program code used to find discrete fractional Fourier transform matrix, Murat Akgül for reading and commenting on the fuzzy clustering section of this thesis, Gün Akkor and Tolga Kartaloğlu for providing some useful programs to my PC which are used in this thesis.

It is a great pleasure to express my special thanks to my husband Aykut for his endless love, support, patience, and tolerance, and to my son Mehmet Fatih who made my life more enjoyable with his father during my Ph. D. studies.

To my husband Aykut and my son Mehmet Fatih

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

Although some sensors provide accurate information on locating and tracking targets, they may not provide identity information (or vice versa), pointing to the need for combining data from multiple sensors using data fusion techniques. The primary aim of data fusion is to combine data from multiple sensors to perform inferences that may not be possible with a single sensor. In robotics applications, data fusion enables intelligent sensing to be incorporated into the overall operation of robots so that they can interact with and operate in unstructured environments without the complete control of a human operator. Data fusion can be accomplished by using geometrically, geographically or physically different sensors at different levels of representation such as signal-, pixel-, feature-, and symbol-level fusion.

Mobile robots need the model of the environment in which they operate for various applications. They can obtain this model partly or entirely using a group of physically identical or different sensors. For instance, considering typical indoor environments, a robot must be able to differentiate planar walls, corners, edges, and cylinders for map-building, navigation, obstacle avoidance, and target-tracking. Reliable differentiation is crucial for robust operation and is highly dependent on the mode(s) of sensing employed.

One of the most useful and cost-effective modes of sensing for mobile robot applications is sonar sensing. The fact that acoustic sensors are light, robust and inexpensive devices has led to their widespread use in applications such as navigation of autonomous vehicles through unstructured environments [1–4], map-building [5–7], target-tracking [8], and obstacle avoidance [9]. Although there are difficulties in the interpretation of sonar data due to poor angular resolution of sonar, multiple and higher-order reflections, and establishing correspondence between multiple echoes on different receivers [10, 11], these difficulties can be overcome by employing accurate physical models for the reflection of sonar.

Sonar ranging systems commonly employ *time-of-flight* (TOF) information, recording the time elapsed between the transmission and reception of a pulse. A comparison of various TOF estimation methods can be found in [12]. Since the standard electronics for the widely-used Polaroid sensor [13] do not provide the echo amplitude directly, most sonar systems rely only on TOF information. Differential TOF models of targets have been used by several researchers: In [14], a single sensor is used for map-building. First, edges are differentiated from planes/corners from a single vantage point. Then, planes and corners are differentiated by scanning from two separate locations using the TOF information in complete sonar scans of the targets. Rough surfaces have been considered in [7, 15]. In [6], a similar approach has been proposed to identify these targets as beacons for mobile robot localization. A tri-aural sensor arrangement which consists of one transmitter and three receivers to differentiate and localize planes, corners, and edges using only the TOF information is proposed in [10]. A similar sensing configuration is used to estimate the radius of curvature of cylinders in [16,17]. Differentiation of planes, corners, and edges is extended to 3-D using three transmitter/receiver pairs (transceivers) in [18, 19] where these transceivers are placed on the corners of an equilateral triangle. Manyika has used differential TOF models for target-tracking [20]. Systems using only qualitative information [8], combining amplitude, energy, and duration of the echo signals together with TOF information [7,21,22], or exploiting the complete echo signal [23] have

also been considered.

Sensory information from a *single* sonar has poor angular resolution and is not sufficient to differentiate the most commonly encountered target primitives [22]. Improved target classification can be achieved by using multi-transducer pulse/echo systems and by employing both amplitude and TOF information. However, a major problem with using the amplitude information of sonar signals is that the amplitude is very sensitive to environmental conditions. For this reason, and also because the standard electronics used in practical work typically provide only TOF data, amplitude information is rarely used. In earlier work, Barshan and Kuc introduce a method based on only amplitude information to differentiate planes and corners [22]. This algorithm is extended to other target primitives in [21] using both amplitude and TOF information. In addition to making use of the amplitude information, the target classification problem is handled more reliably by exploiting the pattern recognition capability of multi-layer neural networks in [24].

In this thesis, information from physically identical sonar sensors located at geographically different sensing sites are combined. Feature-level fusion is used to perform the object recognition task, where additional features can be incorporated as needed to increase the recognition capability of the sensors. Based on the features used, each sensor makes a decision about the type of the target it detects. Due to the uncertainty of the measurements and the multiplicity of decision-makers, conflicts can arise pointing to the need for reliable and robust fusion algorithms. The numerous techniques for fusion can be divided into two categories as parametric and non-parametric. In parametric methods, models of the observations and the fusion process, generally based on the assumption of an underlying probability distribution, are used (i.e., Bayesian methods). In non-parametric methods, assumptions about the underlying probability distributions are not needed, resulting in greater robustness in certain situations (for example, when the noise is non-additive, non-Gaussian or generated by a

nonlinear process).

In this thesis, performances of different classification schemes and fusion techniques in target differentiation and localization of commonly encountered features in indoor robot environments are compared. The classification schemes employed include target differentiation algorithm developed in earlier work [21], statistical pattern recognition techniques which are k-nearest neighbor (k-NN) and generalized k-NN classifiers, kernel estimator, parameterized density estimator and linear discriminant analysis, fuzzy c-means clustering algorithm, and artificial neural networks. The fusion techniques used in this thesis are Dempster-Shafer evidential reasoning, simple majority voting, and different voting schemes with preference ordering and five different reliability measures. These fusion techniques are used based on the target differentiation algorithm of [21]. To the best of our knowledge, a compact, complete and neat comparison of these different approaches supported by experimental verification does not exist for target classification and localization with sonar. The main contribution of this thesis is the comparison of these methods based on experimentally obtained data.

Neural networks have been employed efficiently as pattern classifiers in numerous applications [25]. These classifiers are non-parametric and make weaker assumptions on the shape of the underlying distributions of input data than traditional statistical classifiers. Therefore, they can prove more robust when the underlying statistics are unknown or the data are generated by a nonlinear system. Neural networks have been used in sonar and radar signal processing [26, 27]; for instance, in the identification of ships from observed parametric radar data [28]. The motivation behind the use of neural network classifiers in sonar or radar systems is the desire to emulate the remarkable perception and pattern recognition capabilities of humans and animals, such as the powerful ability of dolphins and bats to extract detailed information about their environments from acoustic echo returns [29–31]. A comparison between neural networks and standard classifiers for radar-specific emitter identification is provided by [32]. An

acoustic imaging system which combines holography with multi-layer feed-forward neural networks for 3-D object recognition is proposed in [33]. A neural network which can recognize 3-D cubes and tetrahedrons independent of their orientation using sonar is described in [34]. Neural networks have also been used in the classification of sonar returns from undersea targets, for example, in [35], where the correct classification percentage of the network employed (90%) exceeds that of a nearest neighborhood classifier (82%). Another application of neural networks to sonar data is in the classification of cylinders under water or in sediment where the targets are made of different materials [29,35], made of the same material but with different diameters [29], or in the presence of a second reflector in the environment [36]. Neural networks have also been used in naval friend-or-foe recognition in underwater sonar [37].

Performance of neural network classifiers is affected by the choice of the parameters of the network structure, training algorithm, and input signals, as well as parameter initialization [38,39]. This thesis also investigates the effect of various representations of input sonar signals and two different training algorithms on the performance of neural networks with different structures used for target classification and localization. The input signals are different functional forms and transformations of amplitude and TOF characteristics of commonly encountered targets acquired by a real sonar system. To the best of our knowledge, these input signals have not been used so far with neural networks in target classification and localization with sonar.

Two non-parametric decision fusion techniques are considered. The first is Dempster-Shafer evidential reasoning which is well-suited for dealing with imprecise evidence and uncertainty in a more rational way than other tools [40–42]. The second technique is majority voting which provides fast and robust fusion in certain problems [43,44]. Despite the fast and robust fusion capability of majority voting, it involves certain consistency problems that limit its usage.

The sensing nodes view the targets at different ranges and angles so that they have

different degrees of reliability. Clearly, proper accounting for these different reliabilities has the potential to considerably improve decision making compared to simple uniform treatment of the sensors. Preference ordering among possible target types and reliability measure assignment is considered, the latter of which essentially amounts to weighting the information from each sensor according to the reliability of that sensor. To the best of our knowledge, the different reliabilities of the sensors have not been exploited so far in sonar sensing, with the sensors being treated uniformly. We compare Dempster-Shafer evidential reasoning and simple and preference-ordered majority voting strategies, both incorporating reliability measures, to identify a strategy that can offer substantial improvement in the classification error.

In this thesis, the *best* classification and localization scheme (which is found to be the neural network classifier trained with the wavelet transformed sonar signals) is applied to map-building for mobile robots. The map of a mobile robot's environment can be provided readily by a human operator to the robot or the robot itself may explore the environment to extract its own map. The second approach is more useful and effective in dynamic environments. The changes in the environment will be sensed by the robot's onboard sensors and suitable updates to the map will be made automatically. Otherwise, the human operator must supply a new map to the robot for every change that occurs in the environment. In most cases, user-supplied map has limited value to the robot since it is difficult for the user to represent the environment in the same level of detail as the capability of the robot's sensors.

There are two commonly used approaches to describe the environment on a map. In the first one, primitive features of the environment such as walls, corners, edges or cylinders and their locations, orientations and sizes are represented (feature-based). In the second, the robot's environment is divided into small regions (usually square shaped) or grids and their occupancy states such as free, occupied or unknown is provided (area-based or grid-based). Area-based maps usually represent the probability of occupancy of

the corresponding subregion, therefore they heavily depend on the probabilistic model of the robot's sensors which results in the requirement of accurate understanding of physics of the corresponding sensors. Due to these limitations of area-based maps, it is more attractive to use feature-based maps in which features are extracted by the cooperation of physically different sensors for more accurate specification of the properties of these features.

How does the robot explore its environment to build its own map? There exists two common strategies: In the first one, the robot explores the environment under the control of a human operator, whereas in the second case the robot uses its own sensors to define an exploring strategy independent of the human operator. Of course, the second approach is more attractive for effective and independent robot operations. In the second case, the most commonly used exploration method by the robot is *wall following* in which the robot follows the walls of the environment while maintaining a fixed distance to the wall. However, the main drawback of this exploring strategy is that if there exists a convex object such as a free-standing pillar in the environment, then the robot will cycle it forever. Moreover, at every step, the robot should take new measurements and correspondence should be established with the previous measurements. Instead of exploring the environment by wall-following, the robot can use some critical points or vantage points to explore. In this thesis, these critical points are the *meet points* which are defined as the points equidistant to three objects in two-dimensional environments. The way to find these points from ultrasonic sensor readings, based on the Generalized Voronoi Diagram, is proposed in [45–47] for a cylindrical robot. After finding these meet points, the environment is scanned at these points and a neural network is used to extract the features of the environment. The correspondence between the features extracted at each meet point is established to obtain a global feature-based map of a mobile robot's environment.

Inclusion of physically different sensors such as infrared sensors and structured-light

systems besides sonars is considered to increase the performance of target classification and localization.

This thesis is organized as follows: basics of sonar sensing and some preliminary work on reliable classification through fusion of the sensors' decisions using Dempster-Shafer evidential reasoning and majority voting is provided in Chapter 2. In Chapter 3, consistency problems of majority voting is addressed and proposed solutions including preference ordering and reliability measures are tested experimentally. In order to find the optimum number of classes existing in sonar data, fuzzy c-means clustering and minimum description length principle are employed in Chapter 4. The effect of various representations of input sonar signals and two different training algorithms on the performance of neural networks with different structures used for target classification and localization is investigated in Chapter 5. In Chapter 6, application of statistical pattern recognition techniques to target classification with sonar is presented. The performances of all classification schemes and fusion methods employed in this thesis for target classification and localization are compared experimentally in a common test pool in Chapter 7. In Chapter 8, an application of the *best* classification and localization scheme to map-building is provided. Physically different sensors besides sonars are included to increase the performance of target classification and localization in Chapter 9. In Chapter 10, concluding remarks are made and directions for future work are discussed.

Chapter 2

PRELIMINARY STUDIES

In this chapter, preliminary studies on target differentiation using sonar for robotics applications are described. The results of these works are used as the building blocks of our thesis work described in the next chapters.

This chapter is organized as follows: Section 2.1 explains the sensing configuration used in this thesis and introduces the target primitives. In Section 2.2, the amplitude and TOF-based classification algorithm is provided as an extension of the amplitude-based plane/corner differentiation algorithm [21,48] developed in the earlier work by Barshan and Kuc [22]. The next section describes Dempster-Shafer belief assignment used by the sensor node and the underlying mass function which is used for target differentiation and Dempster's rule of combination. In Section 2.4, conflict resolution through voting will be highlighted. The applicability of these two methods to our problem is verified by experiments with a practical sonar system in Section 2.5.

2.1 Sonar Sensing

In the commonly used TOF systems, an echo is produced when the transmitted pulse encounters an object and a range measurement $r = ct_\circ/2$ is obtained when the echo amplitude first exceeds a preset threshold level τ back at the receiver at time $t_\circ$. Here, $t_\circ$ is the TOF and c is the speed of sound in air (at room temperature, $c = 343.3$ m/s.). Many ultrasonic transducers operate in this pulse-echo mode [49]. The transducers can function both as receiver and transmitter. Most systems commonly in use are able to detect only the very first echo after pulse transmission.

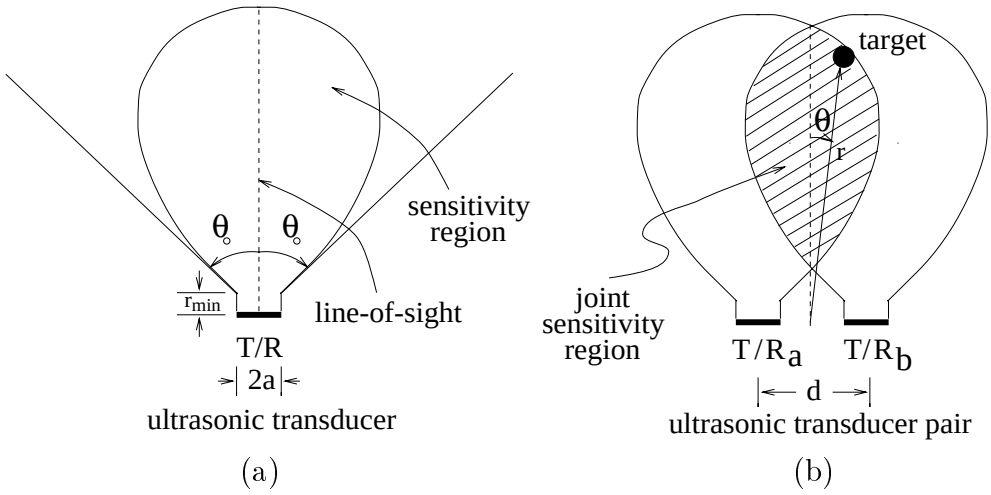


Figure 2.1: (a) Sensitivity region of an ultrasonic transducer. (b) Joint sensitivity region of a pair of ultrasonic transducers. The intersection of the individual sensitivity regions serves as a reasonable approximation to the joint sensitivity region [50].

In this study, the far-field model of a piston-type transducer having a circular aperture is considered [51]. It is observed that the echo amplitude decreases with increasing range r and azimuth θ, which is the deviation angle from normal incidence as illustrated in Figure 2.1(b). The echo amplitude falls below τ when $|\theta| > \theta_\circ$, which is related to the aperture radius a and the resonance frequency $f_\circ$ of the transducer by $\theta_\circ = \sin^{-1}\left(\frac{0.61c}{af_\circ}\right)$ [51]. The radiation pattern is caused by interference effects between different radiating zones on the transducer surface.

The major limitation of ultrasonic transducers comes from their large beamwidth. Although these devices return accurate range data, they cannot provide direct information on the angular position of the object from which the reflection was obtained. The transducer can operate both as transmitter and receiver and detect echo signals reflected from targets within its *sensitivity region* [Figure 2.1(a)]. Thus, with a single stationary transducer, it is not possible to estimate the azimuth of a target with better resolution than the angular resolution of the device which is approximately $2\theta_\circ$. The reflection point on the object can lie anywhere along a circular arc (as wide as the beamwidth) at the measured range. More generally, when one sensor transmits and another receives, both members of the sensor configuration can detect targets located within the *joint sensitivity region*, which is the overlap of the individual sensitivity regions [Figure 2.1(b)]. In this case, the reflection point lies on the arc of an ellipse whose focal points are the transmitting and receiving transducers. The angular extent of these circular and elliptical arcs is determined by the sensitivity regions of the transducers. In our system, two identical acoustic transducers a and b with center-to-center separation d are employed to improve the angular resolution. These two transducers together constitute what we will refer to as a *sensor node* throughout this thesis. The extent of the sensitivity regions is different for different targets which, in general, exhibit different reflection properties. For example, for edge-like or pole-like targets, this region is much smaller but of similar shape, and for planar targets, it is more extended [52].

The target primitives employed in this thesis are *plane*, *corner*, *acute corner*, *edge*, and *cylinder* (Figure 2.2). These target primitives constitute the basic building blocks for most of the surfaces likely to exist in uncluttered robot environments.

Most ultrasonic systems operate below a resonance frequency of 200 kHz so that the propagating waves have wavelengths well above several millimeters. In our case, since the operating wavelength ($\lambda \cong 8.6$ mm at $f_\circ = 40$ kHz) is much larger than the typical roughness of surfaces encountered in laboratory environments, targets in these

environments reflect acoustic beams specularly, like a mirror. Details on the objects which are smaller than the wavelength cannot be resolved [53]. Specular reflections allow the single transmitting-receiving transducer to be viewed as a separate transmitter T and virtual receiver R [4]. Detailed specular reflection models of these target primitives with corresponding echo signal models are provided in [21].

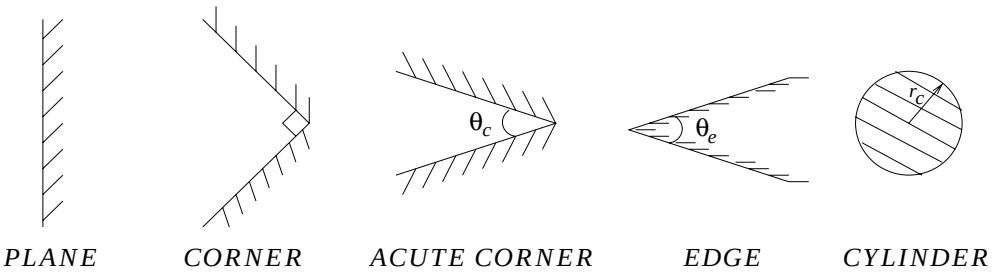


Figure 2.2: Horizontal cross sections of the target primitives modeled and differentiated in this study.

Typical sonar waveforms from a planar target located at $r = 60$ cm and $\theta = 0°$ are given in Figure 2.3. These waveforms are obtained using the sensor configuration illustrated in Figure 2.1(b) with separation $d = 25$ cm. In the figure, A_{aa}, A_{bb}, A_{ab}, and A_{ba} denote the maximum values of the echo signals, and t_{aa}, t_{bb}, t_{ab}, and t_{ba} denote the TOF readings extracted from these signals. The first index in the subscript indicates the transmitting transducer, the second index denotes the receiver. The ideal amplitude and TOF characteristics of these target primitives as a function of the scan angle α are provided in Figures 2.5 and 2.6. The scan angle is the angle between the line corresponding to $\theta = 0°$ and the line-of-sight of the rotating sensor (Figure 2.4).

The characteristics illustrated in Figures 2.5 and 2.6 are obtained by simulating the echo signals according to the models provided in [48]. It can be observed that the echo amplitude decreases with increasing azimuth. In reality, the signals are very noisy and the actual amplitude and TOF data obtained from a real sonar system are far from ideal (Figures 2.7 and 2.8). In these figures, the solid lines correspond to the average over

eight data sets. The level of amplitude and TOF noise is also illustrated by plotting the $\pm 3\sigma_A$ and $\pm 3\sigma_t$ curves together with the average amplitude and TOF curves. Here, σ_A and σ_t are the amplitude and TOF noise standard deviations, respectively. Due to the significant amount of amplitude noise, methods which reduce the resulting uncertainty are needed.

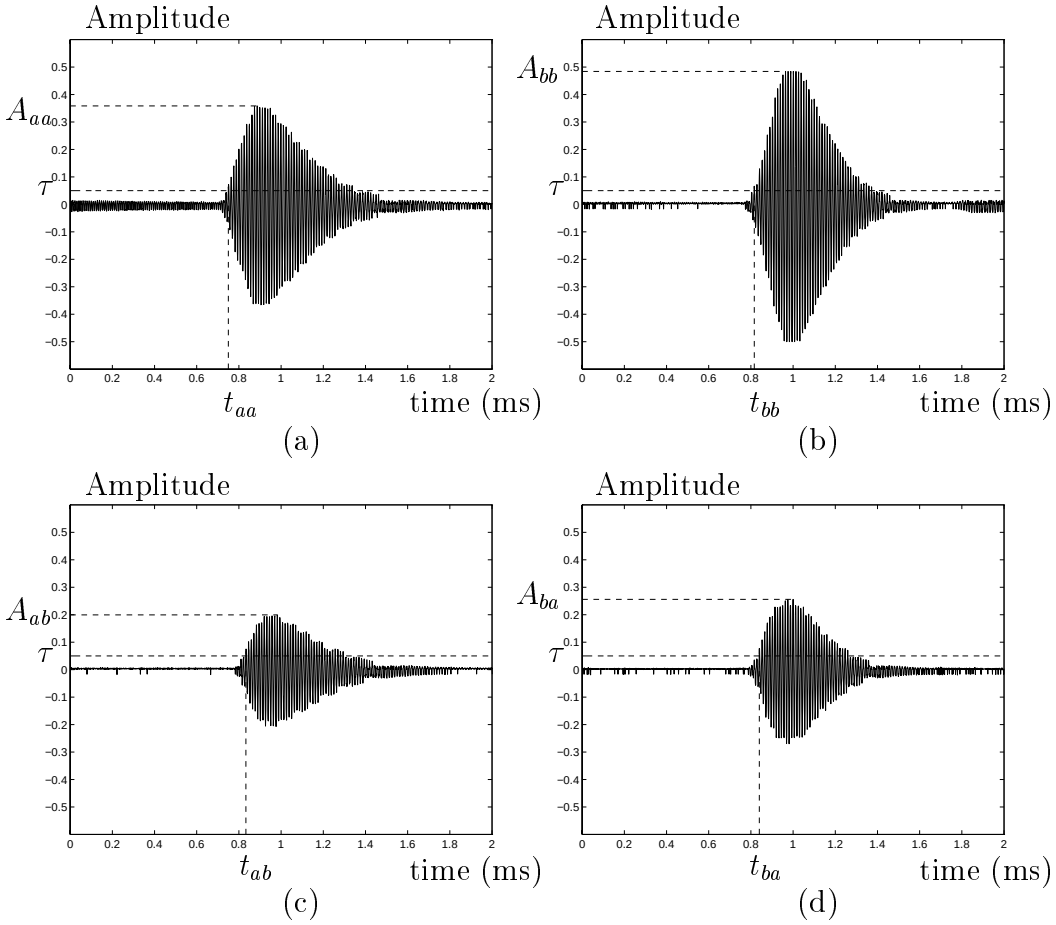


Figure 2.3: Real sonar waveforms obtained from a planar target when (a) transducer a transmits and transducer a receives, (b) transducer b transmits and b receives, (c) transducer a transmits and b receives, (d) transducer b transmits and a receives.

The discrepancy between the real data and the simulations indicates that the models underlying the simulations are far from fully adequate in describing the complexity of

the real situation. In particular, the models do not account for multiple reflections or the possibility of reflections from other objects in the environment. For this reason, we have tested our methods on real data rather than simulations. Nevertheless, the simulations are useful in suggesting qualitative interpretations of the results provided in the next section.

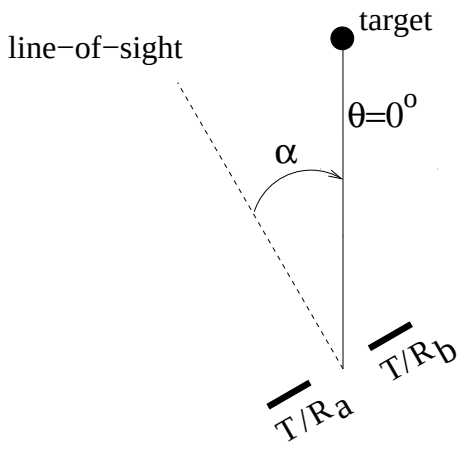


Figure 2.4: Scan angle α.

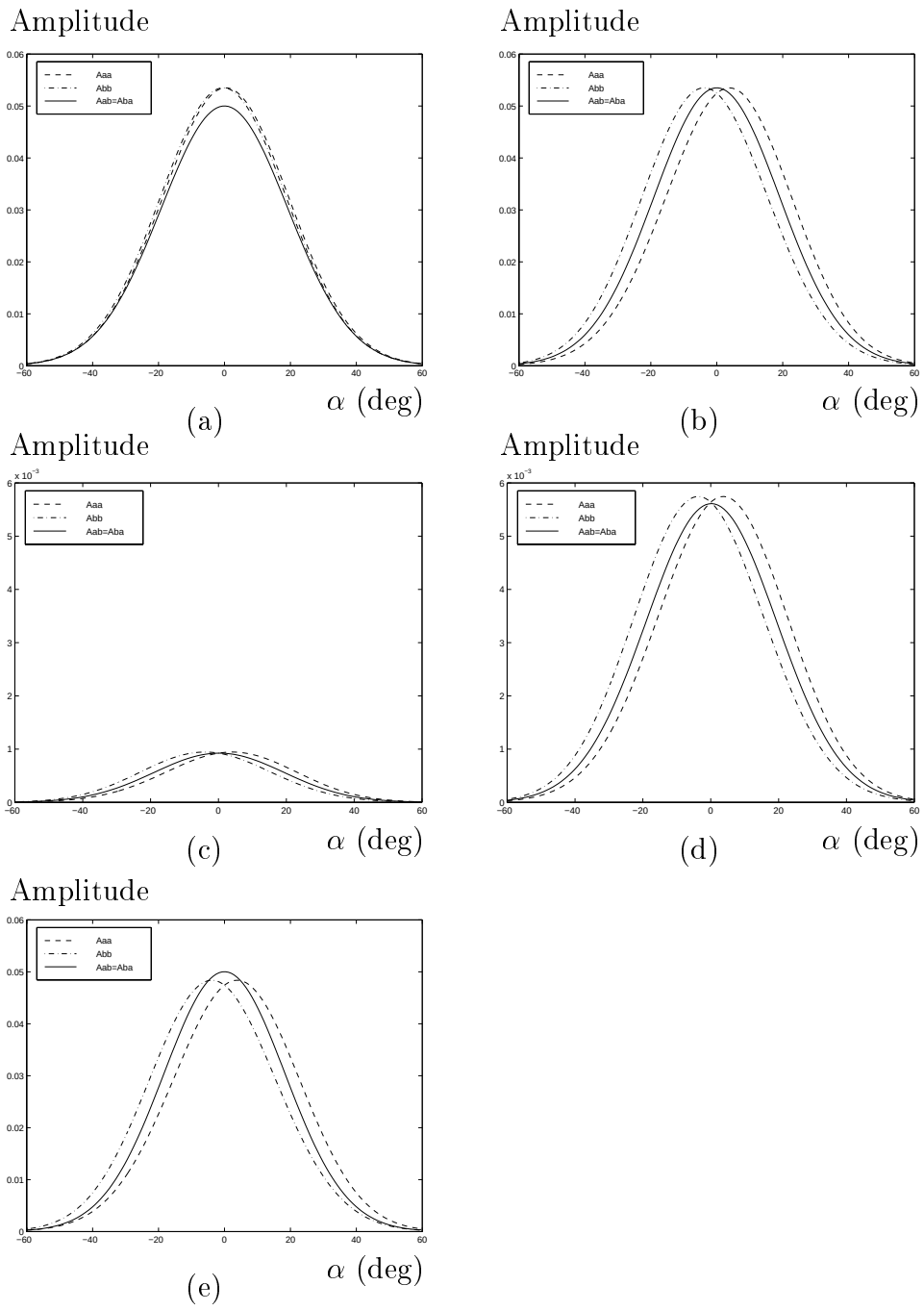


Figure 2.5: Amplitude characteristics at $r = 2$ m for the targets: (a) plane (b) corner (c) edge with $\theta_e = 90°$ (d) cylinder with $r_c = 20$ cm (e) acute corner with $\theta_c = 60°$.

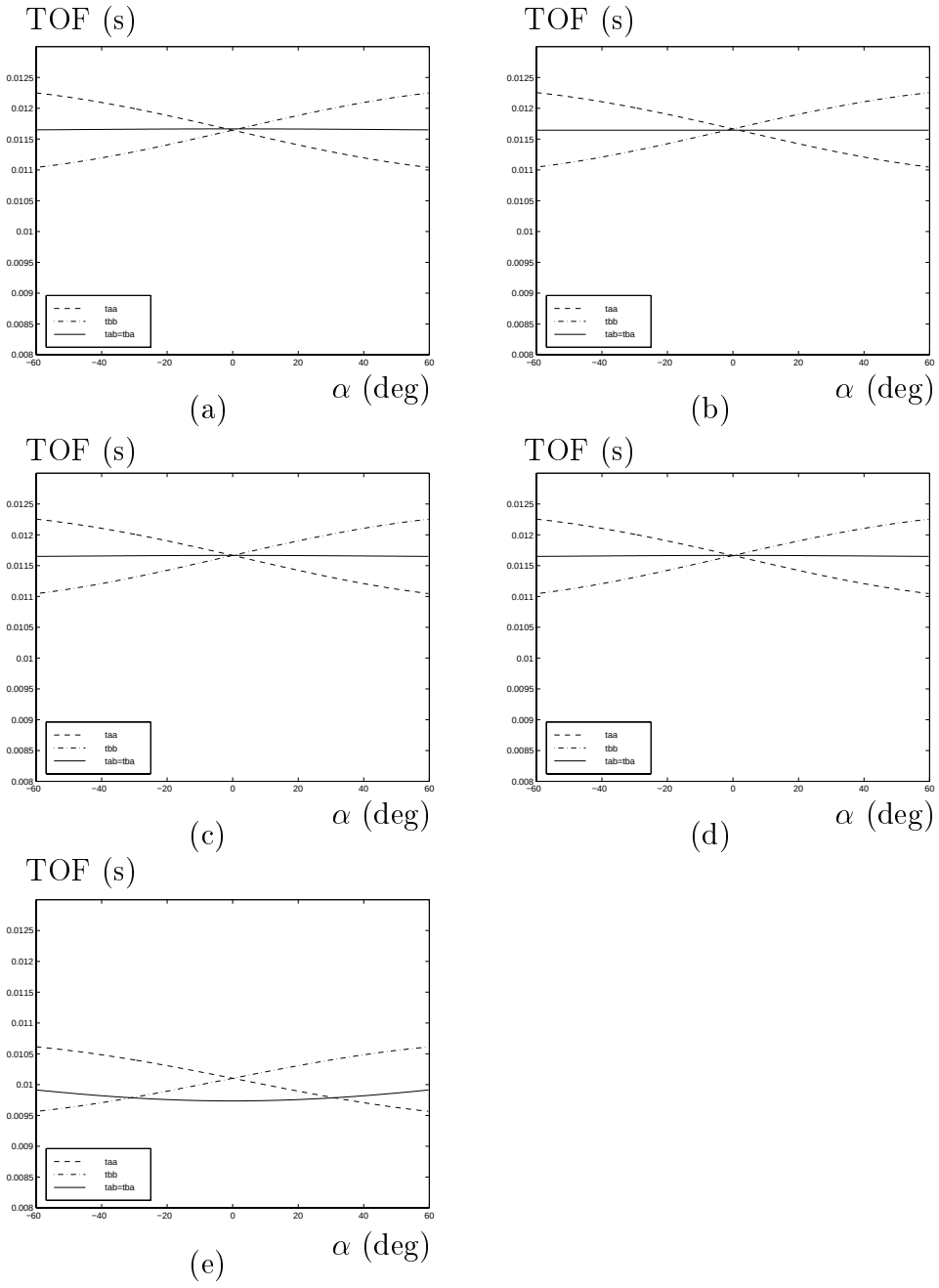


Figure 2.6: TOF characteristics at $r = 2$ m for the targets: (a) plane (b) corner (c) edge with $\theta_e = 90°$ (d) cylinder with $r_c = 20$ cm (e) acute corner with $\theta_c = 60°$.

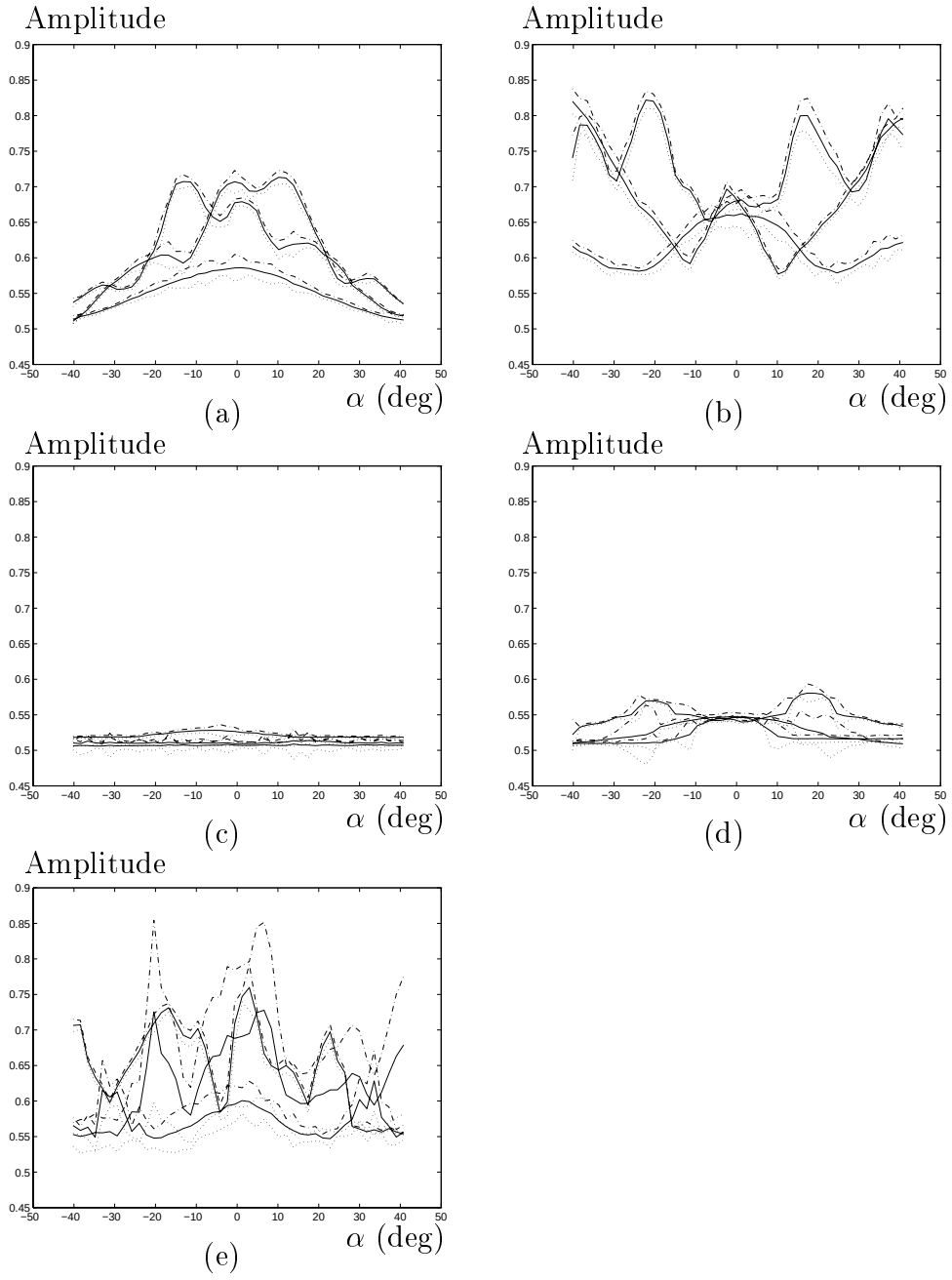


Figure 2.7: Amplitude characteristics which incorporate the amplitude noise ($\pm 3\sigma_A$) for the targets: (a) plane (b) corner (c) edge with $\theta_e = 90°$ (d) cylinder with $r_c = 5$ cm (e) acute corner with $\theta_c = 60°$. Here solid, dashed, and dotted lines correspond to the average over eight data sets, average$+3\sigma_A$ and average$-3\sigma_A$ respectively.

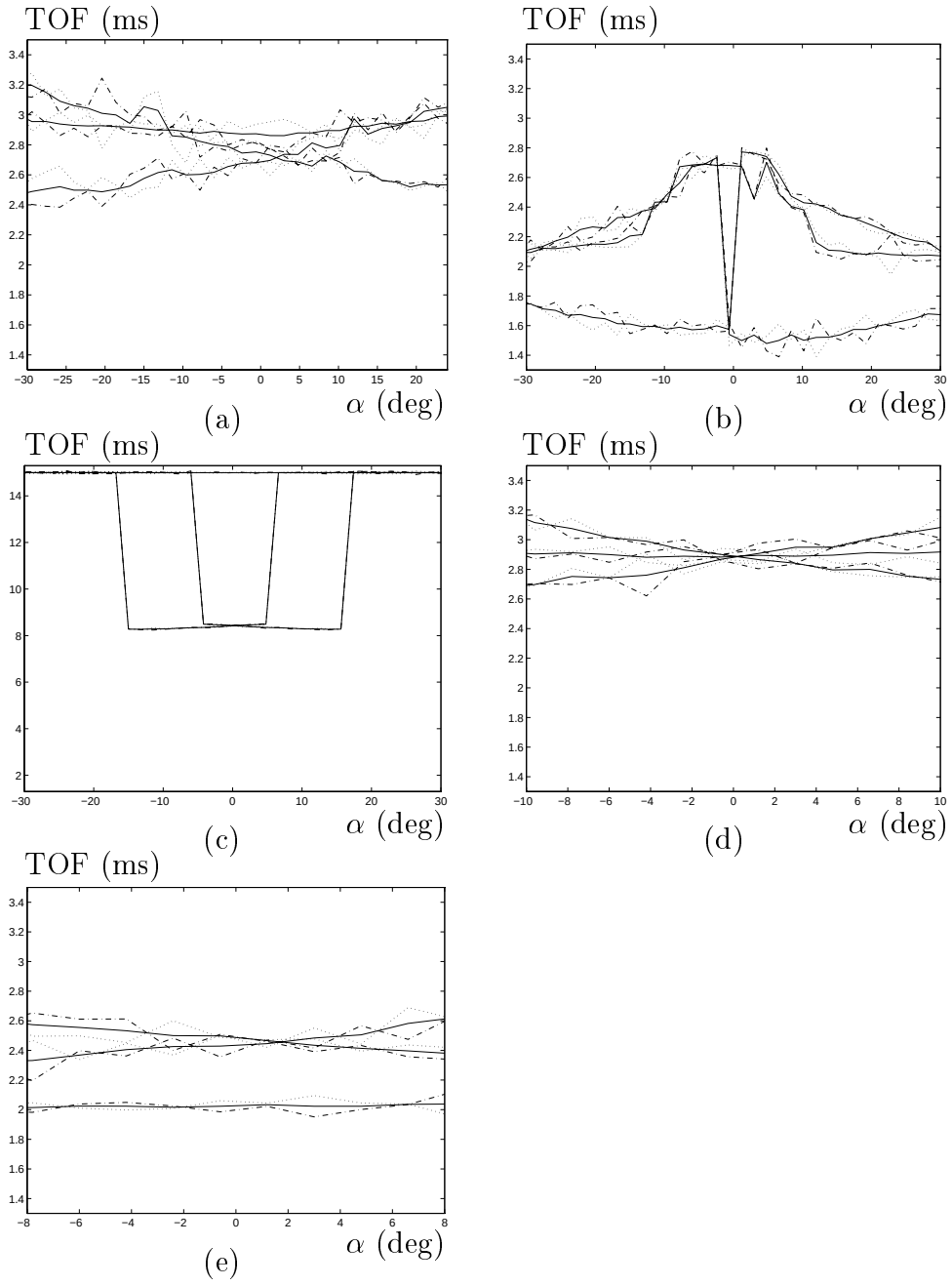


Figure 2.8: TOF characteristics which incorporate the TOF noise ($\pm 3\sigma_t$) for the targets: (a) plane (b) corner (c) edge with $\theta_e = 90°$ (d) cylinder with $r_c = 5$ cm (e) acute corner with $\theta_c = 60°$. Here solid, dashed, and dotted lines correspond to the average over eight data sets, average$+3\sigma_t$ and average$-3\sigma_t$ respectively.

2.2 Target Differentiation Algorithm

In this section, the target differentiation algorithm used in earlier work [21] is summarized. This classification algorithm has its origins in the plane/corner differentiation algorithm developed in another earlier work by Barshan and Kuc [22]. The algorithm of [22] is based on the idea of exploiting amplitude differentials in resolving target type (Figure 2.5). In [21], the algorithm is extended to include other target primitives using both amplitude and TOF differentials based on the characteristics of Figures 2.5 and 2.6. The extended algorithm may be summarized in the form of rules:

if $[t_{aa}(\alpha) - t_{ab}(\alpha)] > k_t\sigma_t$ **and** $[t_{bb}(\alpha) - t_{ba}(\alpha)] > k_t\sigma_t$ then **acute corner** $\rightarrow$ exit

if $[A_{aa}(\alpha) - A_{ab}(\alpha)] > k_A\sigma_A$ **and** $[A_{bb}(\alpha) - A_{ba}(\alpha)] > k_A\sigma_A$ then **plane** $\rightarrow$ exit

if $[\max\{A_{aa}(\alpha)\} - \max\{A_{bb}(\alpha)\}] < k_A\sigma_A$ **and** $[\max\{A_{aa}(\alpha)\} - \max\{A_{ab}(\alpha)\}] < k_A\sigma_A$

then **corner** $\rightarrow$ exit

else **edge**, **cylinder** or **unknown** $\rightarrow$ exit

In the above algorithm, $k_A(k_t)$ is the number of amplitude (TOF) noise standard deviations $\sigma_A(\sigma_t)$ and is employed as a safety margin to achieve robustness in the differentiation process. Differentiation is achievable only in those cases where the difference in amplitudes (TOFs) exceeds $k_A\sigma_A(k_t\sigma_t)$. If this is not the case, a decision cannot be made and the target type remains unknown.

Two variations of this algorithm can be considered: The first takes into account the noise statistics to achieve robustness ($k_A, k_t \neq 0$), whereas the second treats the data as noiseless ($k_A, k_t = 0$). Since the first version is more conservative in decision making, a lower rate of incorrect decisions is expected at the expense of a higher rate of unknown target type. In the second case, there is no safety margin and consequently a larger rate of incorrect decisions and lower rate of unknown target type is expected.

The above algorithm cannot distinguish between edges and cylinders. Referring to Figure 2.5, edges and cylindrical targets can be distinguished only over a small interval

near $\alpha = 0°$. At $\alpha = 0°$, we have $A_{aa}(0) = A_{bb}(0) = A_{ab}(0)$ for an edge, but this equality is not true for a cylinder. Edges and cylinders can be differentiated with a similar configuration of transducers using a method based on radius of curvature estimation [17, 54]. Depending on the radius of the cylinder, it may be possible to differentiate edges and cylinders. An edge is a target with zero radius of curvature. For the cylinder, the radius of curvature has two limits of interest. As $r_c \to 0$ the characteristics of the cylinder approach those of an edge. On the other hand, as $r_c \to \infty$, the characteristics are more similar to those of a plane. By assuming the target is a cylinder first and estimating its radius of curvature [17, 54], it may be possible to distinguish these two targets for relatively large values of r_c.

After determining the target type, range r and azimuth θ for each target can also be estimated from the measurements obtained with the sensor configuration given in Figure 2.1(b). Moreover, wedge angle θ_c of acute corners and radius r_c of cylinders can also be estimated from the sensor measurements [55].

2.3 Dempster-Shafer Evidential Reasoning

In Dempster-Shafer evidential reasoning, each sensor's opinion is tied to a belief measure or basic probability assignment using belief functions [40]. These are set functions which assign numerical degrees of support on the basis of evidence, but also allow for the expression of ignorance: belief can be committed to a set or proposition without commitment to its complement. In the Dempster-Shafer method, *a priori* information is not required and belief assignment is made only when sensor readings provide supportive evidence. Therefore, ignorance can be represented explicitly. Conflict between views is represented by a conflict measure which is used to normalize the sensor belief assignments. In Dempster-Shafer theory, a frame of discernment, Ω, represents a finite universe of propositions and a basic probability assignment, $m(.)$, maps the power set of

Ω to the interval $[0, 1]$. The basic probability mass assignment satisfies the conditions:

$$m(\emptyset) = 0$$
$$\sum_{A \subseteq \Omega} m(A) = 1 \tag{2.1}$$

A set which has a non-zero basic probability assignment is termed a *focal element*.

The belief or total support that is assigned to a set or proposition A is obtained by summing the basic probability assignments over all subsets of A:

$$Bel(A) = \sum_{B \subseteq A} m(B) \tag{2.2}$$

Evidence which does not support A directly does not necessarily support its complement. The plausibility of A, denoted $Pl(A)$, represents evidence which fails to support the negation of A. Dempster-Shafer evidential reasoning has a powerful evidence combination rule called Dempster's rule of combination or Dempster's fusion rule, described later.

In [56], a model of belief functions based on fractal theory is proposed and applied to the classification problem. An extension of Dempster's rule of combination and the belief propagation for a rule-based system which seeks compromise among belief functions is provided in [57]. An alternative rule of combination is provided to eliminate the deficiencies of Dempster's fusion rule from the assumptions on which it is based for robotic navigation [58]. A modified Dempster-Shafer approach, which can take into account the prior information at hand is proposed in [59]. Pattern classification based on the k-nearest neighborhood classifier is addressed from the point of view of Dempster-Shafer theory in [60]. Evidential reasoning theory has also been applied to robotics [58, 61–63] and to model-based failure diagnosis [64]. A comparison of Bayesian and Dempster-Shafer multi-sensor fusion for target identification is provided in [65].

In this study, sensors are assigned beliefs using Dempster-Shafer evidential reasoning and their opinions are combined through Dempster's fusion rule. The assignments for the

target classification problem are made as follows: The uncertainty in the measurements of each sonar pair (sensing node) is represented by a belief function having target type or $feature$ as a focal element with basic probability mass assignment $m(.)$ associated with this feature:

$$BF = \{feature; m(feature)\} \tag{2.3}$$

The mass function is the underlying function for decision making using the Dempster-Shafer method. It is defined based on the algorithm outlined in Section 2.2 and is thus dependent on amplitude and TOF differential signals such that the larger the differential, the larger the degree of belief (see Equations (2.4)–(2.6)). The mass assignment levels are scaled to fall in the interval [0,1]. The basic probability assignment is described below, where $m(p), m(c)$, and $m(ac)$ correspond to plane, corner, and acute corner assignments, respectively:

$$m(p) = (1 - I_4) I_1 \frac{[A_{aa}(\alpha) - A_{ab}(\alpha)] + [A_{bb}(\alpha) - A_{ba}(\alpha)]}{\max[A_{aa}(\alpha) - A_{ab}(\alpha)] + \max[A_{bb}(\alpha) - A_{ba}(\alpha)]} \tag{2.4}$$

$$m(c) = \begin{cases} (1 - I_4) \frac{I_2[A_{ab}(\alpha) - A_{aa}(\alpha)] + I_3[A_{ba}(\alpha) - A_{bb}(\alpha)]}{I_2 \max[A_{ab}(\alpha) - A_{aa}(\alpha)] + I_3 \max[A_{ba}(\alpha) - A_{bb}(\alpha)]} & \text{if } I_2 \neq 0 \text{ or } I_3 \neq 0 \\ \text{else } 0 \end{cases} \tag{2.5}$$

$$m(ac) = I_4 \frac{[t_{aa}(\alpha) - t_{ab}(\alpha)] + [t_{bb}(\alpha) - t_{ba}(\alpha)]}{\max[t_{aa}(\alpha) - t_{ab}(\alpha)] + \max[t_{bb}(\alpha) - t_{ba}(\alpha)]} \tag{2.6}$$

where I_1, I_2, I_3, and I_4 are the indicator functions of the conditions given below:

$$I_1 = \begin{cases} 1 & \text{if } [A_{aa}(\alpha) - A_{ab}(\alpha)] > k_A \sigma_A \textbf{ and } [A_{bb}(\alpha) - A_{ba}(\alpha)] > k_A \sigma_A \\ 0 & \text{otherwise} \end{cases}$$

$$I_2 = \begin{cases} 1 & \text{if } [A_{ab}(\alpha) - A_{aa}(\alpha)] > k_A \sigma_A \\ 0 & \text{otherwise} \end{cases}$$

$$I_3 = \begin{cases} 1 & \text{if } [A_{ba}(\alpha) - A_{bb}(\alpha)] > k_A \sigma_A \\ 0 & \text{otherwise} \end{cases}$$

$$I_4 = \begin{cases} 1 & \text{if } [t_{aa}(\alpha) - t_{ab}(\alpha)] > k_t \sigma_t \textbf{ and } [t_{bb}(\alpha) - t_{ba}(\alpha)] > k_t \sigma_t \\ 0 & \text{otherwise} \end{cases} \tag{2.7}$$

The remaining belief represents ignorance, or undistributed probability mass and is given by

$$m(u) = 1 - [m(p) + m(c) + m(ac)] \tag{2.8}$$

This uncommitted belief is the result of lack of evidence supporting any one target type more than another. The plausibility represents the evidence which fails to support the negation of a target and adds the uncommitted belief to the belief of targets to evaluate maximum possible belief.

Given two independent sources with belief functions

$$\begin{aligned} BF_1 &= \{f_i, m_1(f_i)\}_{i=1}^4 = \{p, c, ac, u; m_1(p), m_1(c), m_1(ac), m_1(u)\} \\ BF_2 &= \{g_j, m_2(g_j)\}_{j=1}^4 = \{p, c, ac, u; m_2(p), m_2(c), m_2(ac), m_2(u)\}, \end{aligned} \tag{2.9}$$

consensus is obtained as the orthogonal sum:

$$\begin{aligned} BF &= BF_1 \oplus BF_2 \\ &= \{h_k, m_c(h_k)\}_{k=1}^4 = \{p, c, ac, u; m_c(p), m_c(c), m_c(ac), m_c(u)\} \end{aligned} \tag{2.10}$$

which is both associative and commutative. The sequential combination of multiple bodies of evidence can be obtained for n sensing nodes as:

$$BF = (((BF_1 \oplus BF_2) \oplus BF_3) \ldots \oplus BF_n) \tag{2.11}$$

Using Dempster's rule of combination:

$$m_c(h_k) = \frac{\sum \sum_{h_k = f_i \cap g_j} m_1(f_i) m_2(g_j)}{1 - \sum \sum_{h_k = f_i \cap g_j = \emptyset} m_1(f_i) m_2(g_j)} \tag{2.12}$$

where $\sum \sum_{h_k = f_i \cap g_j = \emptyset} m_1(f_i) m_2(g_j)$ is a measure of conflict. The consensus belief function representing the feature fusion process has the measures

$$\begin{aligned} m_c(p) &= \frac{m_1(p)m_2(p) + m_1(p)m_2(u) + m_1(u)m_2(p)}{1 - \text{conflict}} \\ m_c(c) &= \frac{m_1(c)m_2(c) + m_1(c)m_2(u) + m_1(u)m_2(c)}{1 - \text{conflict}} \end{aligned}$$

$$
\begin{aligned}
m_c(ac) &= \frac{m_1(ac)m_2(ac) + m_1(ac)m_2(u) + m_1(u)m_2(ac)}{1 - \text{conflict}} \\
m_c(u) &= \frac{m_1(u)m_2(u)}{1 - \text{conflict}}
\end{aligned} \tag{2.13}
$$

In these equations, disagreement between two sensing nodes is represented by the "conflict" term that represents the degree of mismatch in the features perceived at two different sensing sites. The conflict measure is expressed as:

$$
\begin{aligned}
\text{conflict} = {} & m_1(p)m_2(c) + m_1(c)m_2(p) + m_1(p)m_2(ac) \\
& + m_1(ac)m_2(p) + m_1(c)m_2(ac) + m_1(ac)m_2(c)
\end{aligned} \tag{2.14}
$$

After discounting this conflict, the beliefs can be normalized and used in further data fusion operations.

2.4 Conflict Resolution through Voting

Multi-sensor systems exploit sensor diversity to acquire a wider view of a scene or target under observation. This diversity can give rise to conflicts, which must be resolved when the system information is combined to reach a group decision or to form a group value or estimate. The way in which conflict is resolved is encoded in the fusion method.

Non-parametric methods based on voting have been applied widely in reliability problems [66]. A majority voting scheme for fusing features in model-based 3-D object recognition for computer vision systems is presented in [67]. In [68], voting fusion is applied to target detection and compared with Dempster-Shafer evidential reasoning. These two fusion strategies are also compared for pattern classification in [60]. An analysis on the behavior and performance of majority voting in pattern classification is made in [69]. Voting fusion is applied in robotics to determine path of a mobile robot by voting over various possible actions [70]. A voting scheme to improve the task reliability in obstacle avoidance and target-tracking by fusing redundant purposive modules is

proposed in [71]. Combination of voting schemes with prior probabilities which results in maximum likelihood voting is described in [72].

Voting, in its simplest form, has the advantages of being computationally inexpensive and, to a degree, fault-tolerant. In cases where the sensing system itself abstracts the data to make a decision about target type, it may be more efficient to employ the instrument of a vote instead of fine tuning the parametric information. Major drawback of voting is the consistency problem of Arrow which states that there is no voting scheme for selecting from more than two alternatives that is locally consistent under all possible conditions [73].

2.5 Experimental Studies

The two fusion methods were tested on amplitude data acquired in experiments using scanning sonar sensors. The sensor nodes acquire data from scans of a room, making unilateral decisions on target type at each of several viewing angles. These decisions are then fused to reach a group decision.

2.5.1 Experimental Setup

The data were collected at Bilkent University Robotics Research Laboratory, in a small (1.0 m by 1.4 m), rectangular test area created by partitioning off a section of a laboratory. The test area was calibrated by lining the floor space with metric paper, to allow the sensors and targets to be positioned accurately. The room offers an uncluttered environment, with specularly reflecting surfaces. Sensor nodes occupy the fifteen sites shown in Figure 2.8.

The transducers used in our experimental setup are Panasonic transducers that have a much larger beamwidth than the more commonly used Polaroid transducers [13, 74].

The aperture radius of the Panasonic transducer is $a = 0.65$ cm, its resonance frequency is $f_\circ = 40$ kHz, and therefore $\theta_\circ \cong 54^\circ$ for these transducers (Figure 2.1). In the experiments, separate transmitting and receiving elements with a small vertical spacing

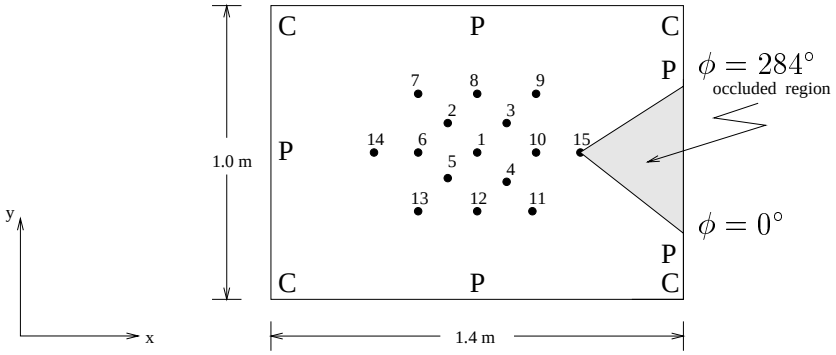


Figure 2.9: The fifteen sensing sites in the rectangular room.

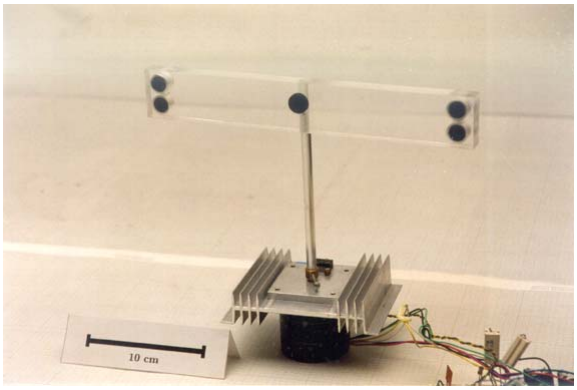


Figure 2.10: Configuration of the Panasonic transducers in the real sonar system. The two transducers on the left collectively constitute one transmitter/receiver. Similarly, those on the right constitute another.

have been used, rather than a single transmitting-receiving transducer. This is because, unlike Polaroid transducers, Panasonic transducers are manufactured as separate transmitting and receiving units (Figure 2.10). The horizontal center-to-center

separation of the transducers used in the experiments is $d = 25$ cm. The entire sensing unit (or the sensor node) is mounted on a small 6 V stepper motor with step size 0.9°. The motion of the stepper motor is controlled through the parallel port of a PC 486 with the aid of a microswitch. Data acquisition from the sonars is through a PC A/D card with 12-bit resolution and 1 MHz sampling frequency. Echo signals are processed on a PC 486 in the C programming language. Starting at the transmit time, 10,000 samples of each echo signal are collected and thresholded to extract the TOF information. The amplitude information is obtained by finding the maximum value of the signal after the threshold is exceeded.

2.5.2 Experimental Results

The two fusion methods are employed with experimental data acquired by the scanning sensing nodes described above. The rules of the target differentiation algorithm summarized in Section 2.2 are taken as the basis in making basic probability mass assignments. Basic probability masses are assigned at each viewing angle ϕ ($0° \leq \phi \leq 284°$) using Equations (2.4)–(2.6). Once the basic probability masses are assigned, the fusion process takes place as follows: In the case of Dempster-Shafer evidential reasoning, Dempster's fusion rule is applied over all the sensing nodes starting with the first one and ending with the last. The target type with maximum belief in the outcome is taken as the decision for a particular viewing angle. In simple majority voting, each sensing node votes for the target for which it has made maximum basic probability mass assignment. The target type receiving the majority of the votes over all sensing nodes is taken as the decision for that viewing angle. To illustrate the accumulation of evidence, the percentage of correct decisions is plotted as a function of the number of sensor nodes used in Figure 2.11. Since the scan step size is 0.9° and the full scan angle is approximately 284°, decisions are made at 315 (= 284/0.9) different viewing angles. When the decisions of fifteen nodes are fused using the Dempster-Shafer method, the

correct decision percentage improves to 86.75%. With simple majority voting, using the same ordering as for the Dempster-Shafer fusion case, the number rises to 87.50%. It can be noted that after simple-voting fusion from about five nodes, the correct decision percentage remains approximately constant around 81%, indicating redundancy in the number of sensor nodes employed. When a single sensor node is used, only about 15% of its decisions are correct. The outstanding 85%, the incorrect and unknown decisions, can be attributed to noise and choice of $k_A(k_t)$ [43].

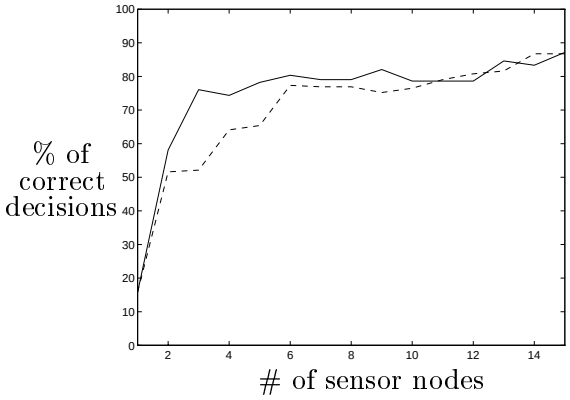


Figure 2.11: Correct decision percentage of Dempster's rule (dashed line) and simple majority voting (solid line) versus the number of sensor nodes employed in the fusion process when an arbitrary order of fusion is used.

The sensor nodes are ordered — essentially, placed in a queue — on the basis of a selected criterion. Fusion occurs in the determined order. Dempster's fusion rule is independent of order [42]. For a fixed group of nodes, the resulting belief will be the same independent of the order in which the beliefs are combined. However, by varying order, it is possible to achieve a preset belief level in a shorter time. The aim is to determine the more informative nodes in the fusion process. Order can also be varied to eliminate redundant, or less informative sensors, thus allowing the preset targeted belief level to be reached using fewer nodes.

The order of combination of beliefs is varied in a number of ways. The fusion order is first generated by taking level of belief as the criterion of node informativeness. The

nodes are placed in order, based on level of belief, irrespective of target type. In making group decisions, the sensor nodes evaluate their decisions relative to those of the group. Starting with the sensor node with the highest belief, nodes are added to the fusion list in the order of smallest distance in belief (highest belief). The objective is to determine whether strength of belief forms a natural selection for sensor nodes. This is analogous to dimensionality reduction in pattern recognition [75], where, among a large number of features, more informative ones, or those with large variances, are selected to improve the efficiency of the classification process. In a similar way, those sensor nodes with larger beliefs are fused first in this study. The objective is to select, from a group of decision makers, a parsimonious set of accurate experts which can achieve a given bound on correct decision rate. A threshold can be set on belief level so that the fusion process is limited to sensors which exceed this level.

The results of maximum-towards-minimum belief fusion are illustrated in Figure 2.12 for the two methods. Here, the fusion process begins with the sensor node which has the highest belief in a target type and continues in the direction of decreasing belief. The performances of the two fusion methods are comparable and the average correct decision percentage is around 85%. The zig-zag pattern in the voting results arises because of switching between odd and even numbers of decision makers.

For comparison purposes, at each viewing angle, fusion was performed only with those five sensor nodes which possess highest belief levels. Clearly, these need not be the same five sensor nodes at each step throughout the scan. With Dempster-Shafer, the results were 84.62% correct on the average, and with voting, 79.91%. Similarly, those five sensor nodes with the lowest belief levels were employed in the fusion process using both methods. In this case, Dempster-Shafer yields only 3.42% correct decisions whereas simple majority voting is 65.81% correct. This significant difference in performance indicates that voting is insensitive to belief levels and more robust than Dempster-Shafer when high uncertainty prevails [76,77]. Since voting emphasizes *numbers* of voters

supporting an outcome, as opposed to the *strength of belief* of voters, which is significant in Dempster fusion, this result is expected.

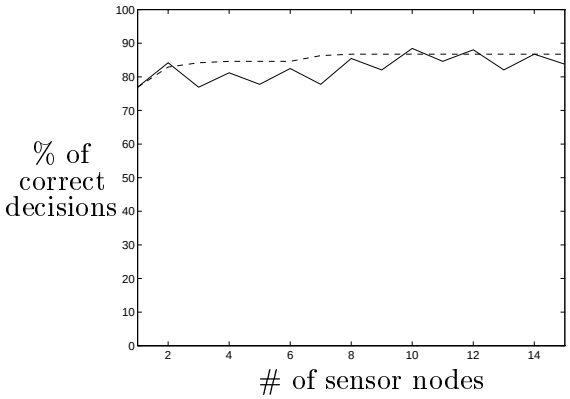


Figure 2.12: Decision fusion from maximum towards minimum belief with Dempster's rule (dashed line) and simple majority voting (solid line) versus the number of sensor nodes employed in the fusion process.

A metric is also defined based on physical distance relative to an arbitrary origin. Starting with a randomly selected node, the beliefs are fused in the order of greatest physical separation. The starting node selects for fusion the node at greatest separation. The next node selected is the one whose distance is greatest from the two nodes which have already combined belief. In this method, the objective is to acquire a comprehensive view of the room more quickly. Distance measures other than physical separation, for example correlation, could be used. Similarly, a minimum distance criterion can be established.

Distance calculations are made as follows: Suppose after fusion over n sensor nodes, the average x and y positions of the group are:

$$x_{av}(n) = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{2.15}$$

$$y_{av}(n) = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{2.16}$$

The $n+1$'th sensor node is chosen such that

$$[x_{n+1} - x_{av}(n)]^2 + [y_{n+1} - y_{av}(n)]^2 \tag{2.17}$$

is maximized (or minimized) over the remaining sensor nodes. In the next step, the new average x and y positions can be found recursively:

$$x_{av}(n+1) = \frac{n}{n+1}\, x_{av}(n) + \frac{1}{n+1}\, x_{n+1} \tag{2.18}$$

$$y_{av}(n+1) = \frac{n}{n+1}\, y_{av}(n) + \frac{1}{n+1}\, y_{n+1} \tag{2.19}$$

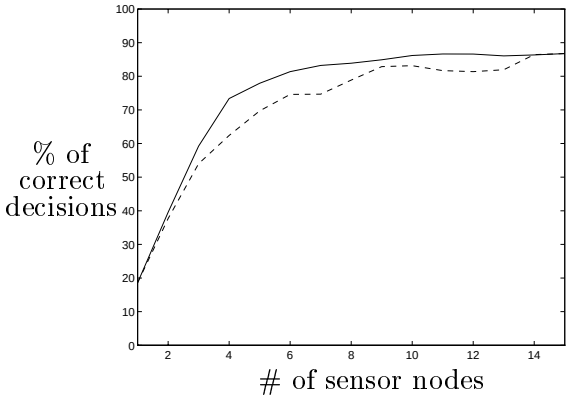


Figure 2.13: Average percentage of correct decisions versus starting sensor node in Dempster's rule in which the decisions of sensor nodes are fused according to maximum distance (solid line) and minimum distance (dashed line).

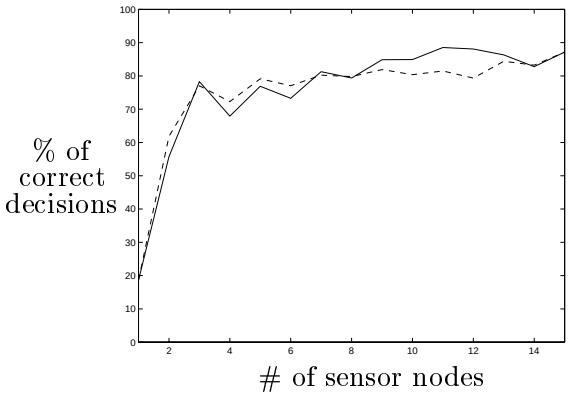


Figure 2.14: Average percentage of correct decisions versus initial sensor node in simple majority voting in which the decisions of the sensor nodes are fused according to maximum distance (solid line) and minimum distance (dashed line).

The results of distance fusion are illustrated in Figures 2.13 and 2.14 for the two fusion methods. In both figures, the results reflect averaging over the fifteen possibilities for

the starting sensor node. In Figure 2.13, Dempster-Shafer fusion results using maximum and minimum distance criteria are compared. Note that performance with all fifteen sensor nodes are identical for maximum and minimum distance fusion as expected: since Dempster-Shafer fusion is commutative and associative, the end result is the same when the same fifteen sensor nodes are used but sorted differently based on minimum and maximum distance criteria. The concern is with intermediate results. In the intermediate fused stages, the maximum performance difference is about 10% between minimum and maximum distance fusion. In Figure 2.14, a similar plot is given for simple majority voting. In this case, note that the average percentage of correct decisions is much larger than the Dempster-Shafer result for up to five or six sensor nodes. The effect of choosing maximum or minimum distance appears to be insignificant for voting. After fusion over six sensor nodes, the performances of the two methods become comparable.

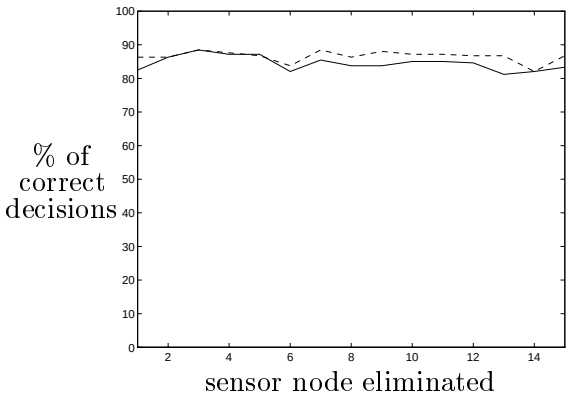


Figure 2.15: Fusion with Dempster's rule (dashed line) and simple majority voting (solid line) versus number of the sensor node which is eliminated in the fusion process.

In Figure 2.15, sensor nodes are eliminated one at a time from the group and the performance of the two methods are compared after fusion over fourteen sensor nodes. The horizontal axis indicates which sensor node is eliminated in the fusion process. From the results, sensor nodes 1, 6, 13 and 14 appear to be most informative. This elimination method can be generalized from individual sensor nodes to groups, to enable the effect of elimination of certain groups of sensor nodes to be studied.

Strategies for grouping sensor nodes during the fusion process are investigated. The sensor nodes are grouped on the basis of the selected criterion. Fusion occurs first within the clusters. In this way, beliefs which support the same criterion are enhanced prior to their fusion with the beliefs of dissenting sensor nodes.

A comparison is made between the simple majority vote outcome and the decision reached when sensor nodes group themselves according to minimum physical distance, fusing only within groups of three. Following this, fusion takes place between the results of each group. This comparative approach investigates the importance of the *numbers* of voters supporting an outcome, which voting emphasizes, as opposed to the *strength of belief* of voters, which is significant in Dempster-Shafer fusion.

Initially, five groups of three sensor nodes are formed based on minimum physical distance as follows: (1-2-3), (4-9-10), (5-6-13), (7-8-11), and (12-14-15). The percentage of correct classification within each group using Dempster-Shafer fusion was 52.14%, 49.57%, 48.29%, 52.56%, and 70.51% respectively. The total correct percentage after fusion over the five groups was 86.75%. Using simple majority voting, the same groups yielded 77.78%, 72.65%, 84.19%, 69.66%, and 91.03% respectively. Taking the majority vote in each group as a vote, the total correct percentage of decisions after voting over the five groups was 89.32%. Note that with voting, success rate of individual groups are much larger compared to Dempster-Shafer. The overall average is also slightly higher.

Further tests were performed, this time using different groups of five sensor nodes. The groups were selected as:
a group of four nodes at minimum distance and one at the furthest distance (2-3-4-5-15);
a line configuration (14-6-1-10-15);
a star configuration (1-14-15-8-12);
(1-2-3-8-15);
(7-2-1-4-11);
(13-5-1-3-9);

(7-1-9-13-11);

(1-2-6-5-15).

The percentages for correct classification within each group using Dempster-Shafer fusion and using voting are shown below:

Sensor grouping	Dempster-Shafer fusion	Voting
(2-3-4-5-15)	63.25 %	76.07 %
(14-6-1-10-15)	81.20 %	89.74 %
(1-14-15-8-12)	87.18 %	91.88 %
(1-2-3-8-15)	58.97 %	82.48 %
(7-2-1-4-11)	71.37 %	76.92 %
(13-5-1-3-9)	63.68 %	84.62 %
(7-1-9-13-11)	69.66 %	80.34 %
(1-2-6-5-15)	76.07 %	85.04 %

In all of these groupings, voting gives a higher total correct decision percentage. The superior performance of voting is partly explained by its relative insensitivity to outliers. Further fusion tests were performed using a group of four sensor nodes which are in agreement. In this case, Dempster-Shafer fusion yields a slightly higher correct decision percentage: 84.62% as compared to 83.33% for voting. At this point, a dissenting sensor node is introduced and fusion performed over the five node values. The voting percentage is stable at 83.33%, but Dempster-Shafer fusion shows a marked decline, from 84.62% to 58.97%. This demonstrates the relative benefit of strength of belief where sensors are in agreement, as opposed to numbers supporting an outcome. For small sets of sensors, unanimity is favored by feature fusion using the Dempster-Shafer method, but the introduction of dissent motivates a more robust approach. Significant improvement in decision accuracy can be achieved using simple majority voting [44]. Moreover, the correct decision percentage of the group (1-14-15-8-12) gives the highest result for both methods which is also higher than the results of 15 sensor nodes. This represents that,

this group of sensor nodes is the more informative than the any other group of sensor nodes employed in this study.

In this chapter, basics of sonar sensing is reviewed and some preliminary work on reliable classification through fusion of the sensors' decisions using Dempster-Shafer evidential reasoning and majority voting is presented. In the next chapter, various solutions to the consistency problem arising in simple majority voting are proposed and these solutions are verified experimentally.

Chapter 3

RELIABILITY MEASURE ASSIGNMENT TO SONAR

In this chapter, consistency problems arising in majority voting are addressed with a view to achieving high classification performance. This is done by introducing preference ordering among the possible target types and assigning reliability measures (which essentially serve as weights) to each decision-making node based on the target range and azimuth estimates it makes and the belief values it assigns to possible target types. The results bring substantial improvement over evidential reasoning and simple majority voting by reducing the target misclassification rate.

This chapter is organized as follows: consistency problems of majority voting and the proposed solutions are summarized in Section 3.1. Assignment of reliability measures to decision-making sonars based on their measurements are discussed in Section 3.2. Section 3.3 describes experimental studies which employ preference ordering and reliability measures to improve the overall performance of majority voting.

3.1 Different Voting Schemes

In simple majority voting, the votes of the different decision makers in the system are given equal weight and the group decision is taken as the outcome with the largest number of votes. Although, simple majority voting provides fast and robust fusion in some problems, there exist some drawbacks that limit its usage. For example, in cases when all outcomes receive equal votes, a group decision cannot be reached. Moreover, it does not take into account whether dissenting classifiers all agree or disagree with each other (i.e., the distribution of the decisions of dissenting classifiers). Consider the following two cases in which fifteen classifiers are employed to classify four target types which are plane (P), corner (C), edge (E) and cylinder (CY):

Case I: Eight classifiers support P
Three classifiers support C
Two classifiers support E
Two classifiers support CY

Case II: Eight classifiers support P
Seven classifiers support C

In both cases, the group decision is plane (P), but are the two decisions equally reliable?

To overcome these drawbacks and to increase the reliability and consistency of the group decision, more sophisticated decision-making schemes can be employed. For this purpose, integer preference orders can be assigned over the possible target types based on the strength of belief. Consider the following situation in which we have three classifiers and four target types, with the preference order given in parentheses:

Classifier 1: P(4) C(3) E(2) CY(1)
Classifier 2: C(4) E(3) CY(2) P(1)

Classifier 3: E(4) CY(3) P(2) C(1)

Note that, in this case, no group decision can be reached by simple majority voting since the first choices of all classifiers are different. However, the total preference order of each target is:

P: 4+1+2=7

C: 3+4+1=8

E: 2+3+4=9

CY: 1+2+3=6

and E wins.

Although this type of approach is more informative, it can also produce conflicting results in some cases. Consider the following situation in which five classifiers are employed to classify the four target types with the preferences given in parentheses:

Classifier 1: P(4) C(3) E(2) CY(1)

Classifier 2: P(4) C(3) CY(2) E(1)

Classifier 3: E(4) P(3) C(2) CY(1)

Classifier 4: C(4) E(3) P(2) CY(1)

Classifier 5: C(4) P(3) CY(2) E(1)

Total preference order of each target type is:

P: 4+4+3+2+3=16

C: 3+3+2+4+4=16

E: 2+1+4+3+1=11

CY: 1+2+1+1+2=7

In this case, the total preference order of plane and corner are equal to each other, again resulting in conflict. To overcome this type of conflict, one can assign reliability measures to the classifiers based on the information at hand. In our case, these classifiers are sonar sensor nodes and apart from target type classification, they can also localize the target based on TOF measurements [48]. Therefore, reliability measures can be assigned based on the location of the target with respect to the sensing node. Assignment of reliability measures will be treated in detail in the next section.

Now, consider the following two cases in which we have reliability values assigned for the five classifiers used in the previous situation:

Case I: Classifier	Reliability
1	0.95
2	0.90
3	0.85
4	0.95
5	0.90

The total preference order of each target type are:

$$\begin{aligned}
\text{P: } 0.95 \times 4 + 0.90 \times 4 + 0.85 \times 3 + 0.95 \times 2 + 0.90 \times 3 &= 14.55 \\
\text{C: } 0.95 \times 3 + 0.90 \times 3 + 0.85 \times 2 + 0.95 \times 4 + 0.90 \times 4 &= 14.65 \\
\text{E: } 0.95 \times 2 + 0.90 \times 1 + 0.85 \times 4 + 0.95 \times 3 + 0.90 \times 1 &= 9.95 \\
\text{CY: } 0.95 \times 1 + 0.90 \times 2 + 0.85 \times 1 + 0.95 \times 1 + 0.90 \times 2 &= 6.35
\end{aligned}$$

Then, C wins.

Now, consider the case where the reliability of classifier 4 is reduced from 0.95 to 0.85:

Case II: Classifier	Reliability
1	0.95

2	0.90
3	0.85
4	0.85
5	0.90

The total preference orders of each target type are:

$$\begin{aligned}
\text{P: } 0.95 \times 4 + 0.90 \times 4 + 0.85 \times 3 + 0.85 \times 2 + 0.90 \times 3 &= 14.35 \\
\text{C: } 0.95 \times 3 + 0.90 \times 3 + 0.85 \times 2 + 0.85 \times 4 + 0.90 \times 4 &= 14.25 \\
\text{E: } 0.95 \times 2 + 0.90 \times 1 + 0.85 \times 4 + 0.85 \times 3 + 0.90 \times 1 &= 9.65 \\
\text{CY: } 0.95 \times 1 + 0.90 \times 2 + 0.85 \times 1 + 0.85 \times 1 + 0.90 \times 2 &= 6.25
\end{aligned}$$

Then, P wins.

Note that the slight change in the reliability of classifier 4 is sufficient to reach a different group decision. Reliability measure assignment needs closer examination since reliability measures more suitable to real situations are likely to result in more accurate group decisions.

3.2 Reliability Measure Assignment

In this section, a description of the assignment of different reliability measures to the sensing nodes based on their current range and azimuth estimates and their belief assignment to target types is given.

Assignment of belief to range and azimuth estimates is based on the simple observation that the closer the target is to the surface of the transducer, the more accurate is the range reading, and the closer the target is to the line-of-sight of the transducer, the more accurate is the azimuth estimate [52]. This is due to the physical properties of sonar: signal amplitude decreases with r and with $|\theta|$. At large ranges and

large angular deviations from the line-of-sight, signal-to-noise ratio is smaller. Most accurate measurements are obtained along the line-of-sight ($\theta = 0^{\circ}$) and at close proximity to the sensor node. Therefore, belief assignments to range and azimuth estimates derived from the TOF measurements can be made as follows:

$$m(r) = \frac{r_{max} - r}{r_{max} - r_{min}} \tag{3.1}$$

$$m(\theta) = \frac{\theta_{\circ} - |\theta|}{\theta_{\circ}} \tag{3.2}$$

Note that, belief of r takes its maximum value of one when $r = r_{min}$ and its minimum value of zero when $r = r_{max}$. Similarly, belief of θ is one when $\theta = 0^{\circ}$ and zero when $\theta = \pm\theta_{\circ}$.

The four different reliability measures assigned to sensor node i are different combinations of the range and azimuth belief functions:

$$\begin{aligned} rel_i^1 &= m(r_i)m(\theta_i) \\ rel_i^2 &= \min\{m(r_i), m(\theta_i)\} \\ rel_i^3 &= \frac{m(r_i) + m(\theta_i)}{2} \\ rel_i^4 &= \max\{m(r_i), m(\theta_i)\} \end{aligned} \tag{3.3}$$

In these equations, each reliability measure takes values in the interval $[0, 1]$. Here, a reliability measure of one corresponds to a maximally reliable sensing node, whereas a reliability measure of zero represents a totally unreliable sensing node. Moreover, their relative magnitudes can be ordered as $rel_i^1 \leq rel_i^2 \leq rel_i^3 \leq rel_i^4$. According to this inequality, rel_i^4 is the more optimistic measure whereas rel_i^1 is the more pessimistic one. Another alternative is to set the reliability measure proportional to the difference between belief values assigned to the first two preferences of each sensing node as an indicator of how strongly that sensing node believes in its first choice. This way, the distribution of the belief values assigned to different target types is partially taken into account. Hence, the fifth reliability measure assignment can be made as follows:

$$rel_i^5 = m(\text{first choice}) - m(\text{second choice}) \tag{3.4}$$

These reliability measures have also been incorporated into Dempster-Shafer evidential reasoning by multiplying Equations (2.4)–(2.6) by the reliability rel_i of a particular sensor node and finding the uncommitted belief by $m_i'(u) = 1 - rel_i[m_i(p) + m_i(c) + m_i(ac)]$. The effect of these different reliability measures on the classification performance of majority voting and evidential reasoning is presented in the next section.

3.3 Experimental Studies

In this section, we describe the experimental procedures used in comparing the various fusion methods described above.

3.3.1 Experimental Setup

The data were collected at Bilkent University Robotics Research Laboratory, in five small experimental test areas created by partitioning off sections of a laboratory. The rooms offer an uncluttered environment, with specularly reflecting surfaces. The number of sensing nodes used were 15, 8, 4, 9, 7 in the rooms shown in Figure 3.1. The first room (*Room A*) which is the same rectangular room used in Section 2.5, consists of only the targets (planes and corners) that can be differentiated by the algorithm summarized in Section 2.2 (Figure 3.1(a)). In addition to planes and corners, the second, third, and fourth rooms (*Rooms B, C,* and *D*) contain edges that cannot be differentiated by this algorithm (Figure 3.1(b) and (c)). In *Rooms D* and *E*, cylindrical targets are also present in the environment. The same sensor node described in Section 2.5 is used to collect the amplitude and TOF data.

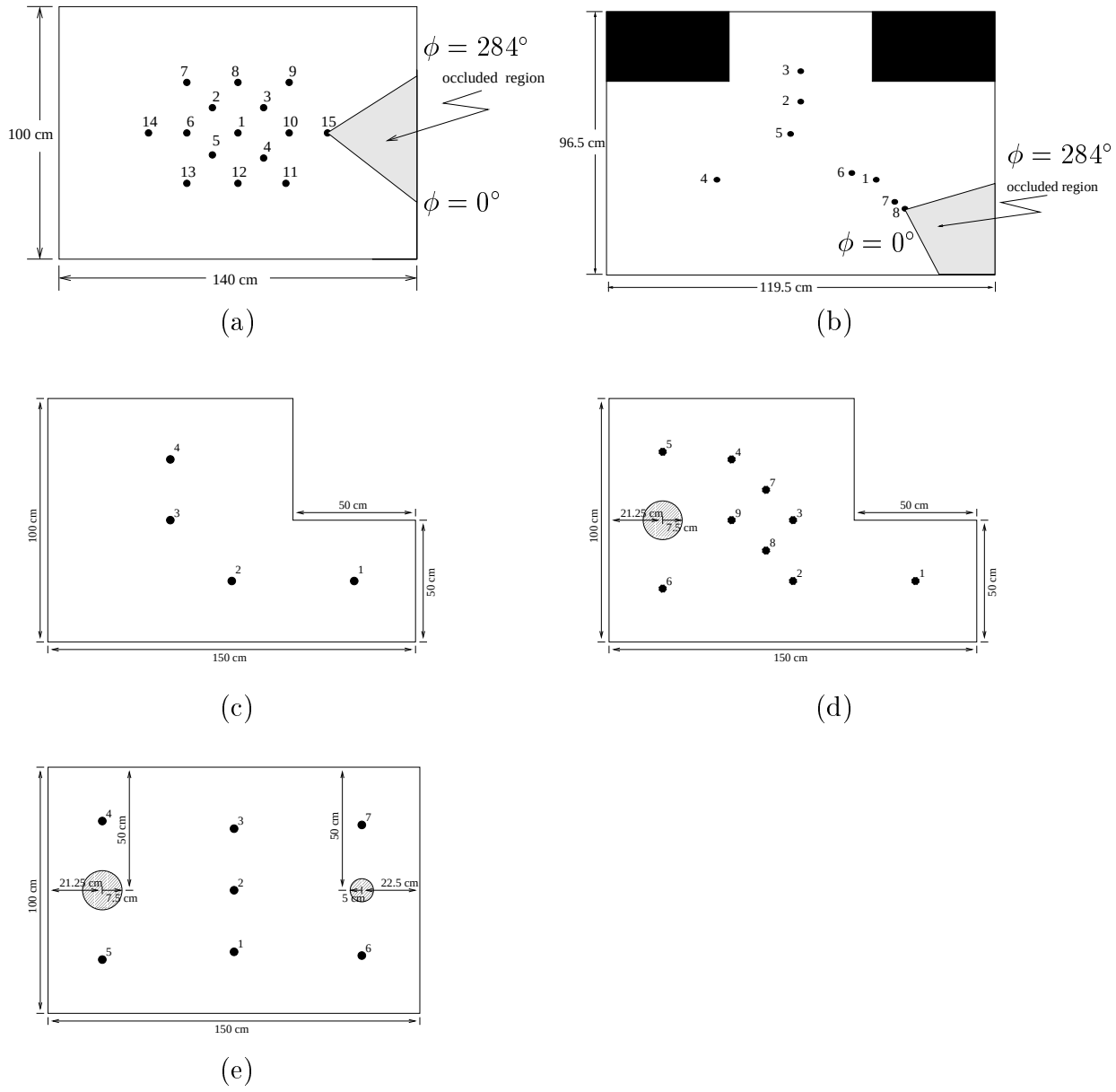


Figure 3.1: Experimental test rooms (a) *Room A*, (b) *Room B*, (c) *Room C*, (d) *Room D*, and (e) *Room E*.

3.3.2 Experimental Results

The sensor beliefs are fused using simple majority voting and Dempster's rule of combination. The two fusion methods in their simple form and when reliability measures

are incorporated are tested with experimental data acquired by the scanning sensing nodes described above. The opinions of all sensor nodes in each room are combined, starting with the first sensing node in that room and ending with the last one. To illustrate the accumulation of evidence, Figure 3.2 shows the percentage of correct classification as a function of the number of sensing nodes used in *Rooms A* and *B*.

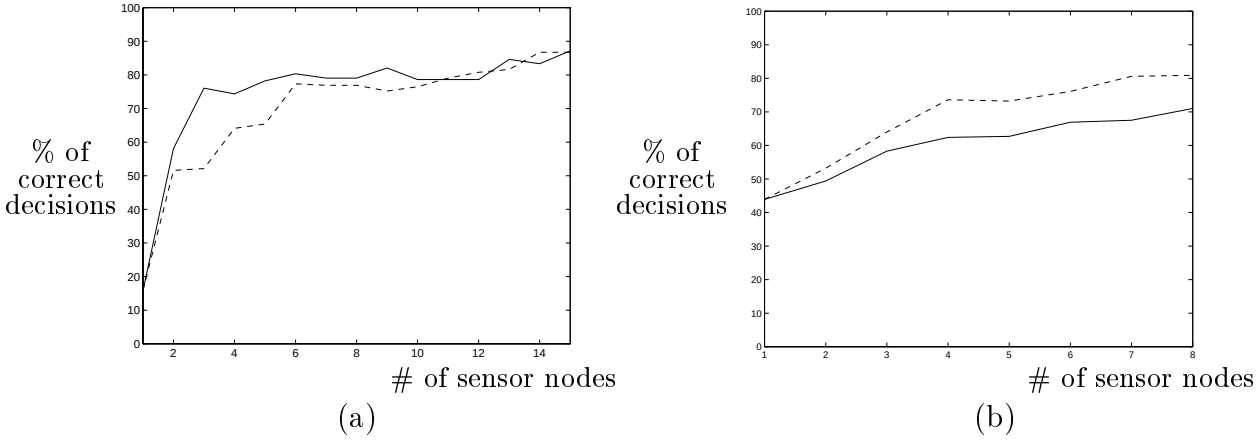


Figure 3.2: Correct decision percentage of Dempster's rule (dashed line) and simple majority voting (solid line) versus the number of sensor nodes employed in the fusion process when an arbitrary order of fusion is used for (a) *Room A* (Figure 2.11) (b) *Room B*.

When a single sensing node is employed and the average of the correct decision percentages is taken over all five rooms, only about 30.6% of the decisions are correct. The outstanding 69.4% incorrect decisions can be attributed to noise and the choice of $k_A(k_t)$. When the decisions of all nodes are fused using Dempster-Shafer and majority voting methods in their simple form, the average correct decision percentage improves to 74.4% and 68.5%, respectively. In *Room A*, simple majority voting outperforms Dempster's rule of combination up to 10 sensing nodes; after this number, performances of the two methods become comparable. However, when targets that cannot be classified by the differentiation algorithm are included in the environment (as in *Rooms B, C, D, E*), Dempster's rule of combination outperforms simple majority voting for any number of sensing nodes used. These results indicate that Dempster-Shafer method in its simple form can handle imprecise evidence more reliably than simple majority voting.

To further improve the target classification performance, preference ordering with and without reliability measures is incorporated in majority voting, and reliability measures are incorporated in Dempster-Shafer evidential reasoning. Preference ordering is considered in two different ways: In the first case, preference orders are taken as integers between 1–4, where the larger the value of the integer, the higher is the preference for that target type. In the second case, the preference orders are taken to be the belief values assigned to each target type. It was observed that the second choice always resulted in higher percentage of correct decisions. Therefore, only the percentages of correct decisions for the second case using various reliability measures are tabulated in Tables 3.2, 3.4, 3.6, 3.8, and 3.10. From these tables, it can be observed that incorporating preference ordering in majority voting without reliability measures (i.e., $rel_i = 1$) already improves on the results obtained with simple majority voting.

With both fusion methods, inclusion of reliability measures brings further improvement compared to using their simple forms. Majority voting with reliability measures and preference ordering performs better than Dempster-Shafer method with reliability measures. When the averages of the best results over the five rooms is taken, the results obtained using Dempster-Shafer and majority voting methods with reliability measures are 77.6% and 81.2%, respectively.

For example, in *Room A* (Tables 3.1 and 3.2), the correct decision percentage achieved with majority voting with preference ordering using the fifth reliability measure (95.1%) is higher than the result obtained with Dempster's rule using the same reliability measure (90.6%). For simple majority voting and simple Dempster-Shafer method, these numbers are 87.5% and 86.8%, and the improvement in the classification error is by a factor of 2.6 and 1.4, respectively.

In *Room B* (Tables 3.3 and 3.4), the highest correct decision percentage achieved with majority voting with preference ordering using the third reliability measure (84.4%) is higher than the best result obtained with Dempster-Shafer method using

the fifth reliability measure (83.8%). For simple majority voting and simple Dempster-Shafer method, these numbers are 71.0% and 80.9%, and the improvement in the misclassification rate is by a factor of 1.9 and 1.2, respectively. These results indicate that majority voting with reliability measures and preference ordering can deal with imprecise evidence in a more reliable way than evidential reasoning with reliability measures.

Although the percentages of correct decisions obtained with the different reliability measures are comparable, among the five reliability measures, rel_i^5 results in slightly better classification rate on the average. This is usually followed by rel_i^3. For example, in *Room C*, after the decisions of all sensing nodes are fused, the fifth reliability measure gives the highest percentage of correct differentiation with Dempster-Shafer method and is followed by the third, fourth, second, and first measures. With majority voting, the fifth and fourth measures give equal results, followed by the third, second, and first measures.

# of nodes used	DS ($rel_i = 1$)	DS with reliability measures				
		rel_i^1	rel_i^2	rel_i^3	rel_i^4	rel_i^5
1	15.8	15.3	15.3	15.8	15.8	15.8
2	38.5	40.8	41.5	44.3	39.6	45.5
3	52.1	56.4	54.9	57.5	54.7	58.5
4	64.1	63.9	63.9	66.2	64.1	65.1
5	65.4	65.8	65.4	68.2	64.5	69.1
6	77.4	77.0	77.8	77.5	76.1	77.5
7	76.9	77.3	77.9	77.8	76.5	78.2
8	76.9	79.7	80.1	79.7	76.9	79.4
9	75.2	79.2	79.9	79.5	75.2	79.3
10	76.5	81.4	82.0	80.8	77.8	82.1
11	79.1	82.6	83.7	81.2	82.1	84.2
12	80.8	81.2	81.2	82.9	82.1	85.0
13	81.6	82.5	82.5	83.3	82.9	87.6
14	86.8	88.9	88.9	89.7	86.8	90.6
15	86.8	89.7	90.2	89.7	86.8	90.6

Table 3.1: Correct decision percentages of Dempster-Shafer method (DS) without/with reliability measures in *Room A*.

# of nodes used	SMV	MV with preference ordering					
		$rel_i = 1$	rel_i^1	rel_i^2	rel_i^3	rel_i^4	rel_i^5
1	15.8	15.8	15.3	15.3	15.8	15.8	15.8
2	64.5	74.9	71.9	71.9	74.9	74.9	74.9
3	77.8	86.1	84.5	84.5	86.5	86.1	87.1
4	76.1	82.5	81.3	81.3	82.7	82.4	83.1
5	78.2	84.3	83.3	83.3	84.3	84.2	85.0
6	80.3	84.9	84.6	84.6	85.2	84.8	85.5
7	79.1	83.4	83.4	83.4	83.7	83.4	83.8
8	79.1	83.0	83.3	83.3	83.3	83.0	83.4
9	82.1	91.3	93.7	93.5	94.5	91.5	94.5
10	78.6	87.2	89.5	89.5	88.0	87.0	88.7
11	78.6	86.1	86.6	86.8	85.7	85.9	88.3
12	78.6	85.2	85.0	85.0	85.7	84.8	86.9
13	84.6	91.1	91.1	91.3	91.7	91.1	93.8
14	83.3	89.7	91.2	91.2	91.4	89.7	91.9
15	87.5	93.0	94.9	94.9	94.7	93.8	95.1

Table 3.2: Correct decision percentages of simple majority voting (SMV), and majority voting (MV) schemes employing preference ordering without/with reliability measures in *Room A*.

# of nodes used	DS ($rel_i = 1$)	DS with reliability measures				
		rel_i^1	rel_i^2	rel_i^3	rel_i^4	rel_i^5
1	43.9	41.1	41.1	43.9	43.9	43.9
2	53.2	56.2	56.7	57.7	59.5	58.4
3	64.0	65.1	65.1	66.4	68.0	69.1
4	73.6	73.9	74.2	75.4	75.5	76.6
5	73.2	74.5	74.2	76.4	77.2	79.0
6	76.1	76.4	76.4	78.7	78.8	80.3
7	80.6	80.7	80.7	81.4	81.5	82.8
8	80.9	81.1	81.1	82.5	82.5	83.8

Table 3.3: Correct decision percentages of Dempster-Shafer method (DS) without/with reliability measures in *Room B*.

# of nodes used	SMV	MV with preference ordering					
		$rel_i = 1$	rel_i^1	rel_i^2	rel_i^3	rel_i^4	rel_i^5
1	43.9	43.9	41.1	41.1	43.9	43.9	43.9
2	49.4	76.8	65.0	65.0	72.3	73.9	69.1
3	58.3	79.0	74.5	74.8	79.0	79.6	73.2
4	62.4	83.1	77.7	78.0	85.4	84.7	81.2
5	62.7	81.8	79.0	79.0	82.8	82.5	79.6
6	66.9	81.5	79.6	79.6	81.8	81.8	80.9
7	67.5	83.4	81.2	81.5	82.2	82.2	83.1
8	71.0	79.6	81.5	81.8	84.4	83.8	84.1

Table 3.4: Correct decision percentages of simple majority voting (SMV), and majority voting (MV) schemes employing preference ordering without/with reliability measures in *Room B*.

# of nodes used	DS ($rel_i = 1$)	DS with reliability measures				
		rel_i^1	rel_i^2	rel_i^3	rel_i^4	rel_i^5
1	31.1	30.6	30.6	31.1	31.1	31.1
2	35.0	37.7	38.5	42.2	40.0	42.9
3	50.8	54.2	54.8	57.7	55.2	57.3
4	63.9	64.2	64.7	66.0	65.3	66.2

Table 3.5: Correct decision percentages of Dempster-Shafer method (DS) without/with reliability measures in *Room C*.

# of nodes used	SMV	MV with preference ordering					
		$rel_i = 1$	rel_i^1	rel_i^2	rel_i^3	rel_i^4	rel_i^5
1	31.1	31.1	30.6	30.6	31.1	31.1	31.1
2	31.7	44.4	38.6	38.6	40.5	40.5	41.6
3	42.1	51.5	47.4	47.4	52.1	53.8	53.2
4	51.9	66.7	67.4	67.4	68.9	69.4	69.4

Table 3.6: Correct decision percentages of simple majority voting (SMV) and majority voting (MV) schemes employing preference ordering without/with reliability measures in *Room C*.

# of nodes used	DS ($rel_i = 1$)	DS with reliability measures				
		rel_i^1	rel_i^2	rel_i^3	rel_i^4	rel_i^5
1	37.4	36.7	36.7	37.4	37.4	37.4
2	53.4	55.0	54.9	56.3	56.2	56.6
3	58.6	59.2	59.5	61.8	61.5	61.5
4	59.5	61.2	61.5	67.8	67.6	66.6
5	61.3	65.6	65.8	69.5	69.3	68.8
6	66.4	69.9	70.0	70.7	70.7	70.1
7	68.7	71.3	71.4	72.0	72.6	72.1
8	69.3	71.6	71.7	73.2	73.3	72.6
9	71.3	71.9	72.0	74.7	74.7	73.6

Table 3.7: Correct decision percentages of Dempster-Shafer method (DS) without/with reliability measures in *Room D*.

# of nodes used	SMV	MV with preference ordering					
		$rel_i = 1$	rel_i^1	rel_i^2	rel_i^3	rel_i^4	rel_i^5
1	37.4	37.4	36.7	36.7	37.4	37.4	37.4
2	48.3	55.5	50.4	50.4	55.3	55.2	54.6
3	52.9	65.1	62.7	62.7	66.5	66.8	66.6
4	61.5	68.1	67.2	67.4	70.5	71.6	70.8
5	59.5	73.7	72.0	72.0	74.5	76.8	76.5
6	61.3	74.5	74.5	74.9	76.8	77.8	77.9
7	66.4	74.8	75.2	75.7	78.2	78.9	79.2
8	67.0	75.9	76.2	76.8	79.3	79.3	79.5
9	67.6	76.0	78.0	78.3	81.5	81.2	81.1

Table 3.8: Correct decision percentages of simple majority voting (SMV) and majority voting (MV) schemes employing preference ordering without/with reliability measures in *Room D*.

# of nodes used	DS ($rel_i = 1$)	DS with reliability measures				
		rel_i^1	rel_i^2	rel_i^3	rel_i^4	rel_i^5
1	24.5	22.4	22.4	24.5	24.5	24.5
2	42.4	44.0	44.0	46.4	47.1	46.7
3	49.5	54.1	53.6	56.5	56.0	56.0
4	57.1	58.7	58.7	63.6	63.6	60.1
5	61.4	61.9	62.2	66.3	65.2	66.6
6	68.5	69.0	69.5	71.2	70.1	71.4
7	69.0	69.8	70.1	72.8	71.7	72.0

Table 3.9: Correct decision percentages of Dempster-Shafer method (DS) without/with reliability measures in *Room E*.

# of nodes used	SMV	MV with preference ordering					
		$rel_i = 1$	rel_i^1	rel_i^2	rel_i^3	rel_i^4	rel_i^5
1	24.5	24.5	22.4	22.4	24.5	24.5	24.5
2	37.5	53.6	40.8	40.8	47.1	46.7	46.7
3	38.6	56.0	54.6	54.0	57.2	57.6	52.0
4	48.9	61.5	56.5	56.5	62.3	65.2	64.0
5	53.3	65.3	63.2	63.2	69.9	69.0	64.0
6	60.9	68.6	69.7	69.7	72.7	72.3	68.9
7	64.7	68.0	70.8	70.8	75.4	75.0	75.5

Table 3.10: Correct decision percentages of simple majority voting (SMV) and majority voting (MV) schemes employing preference ordering without/with reliability measures in *Room E*.

In this chapter, consistency problems arising in majority voting have been addressed. Various solutions including preference ordering among the possible target types and assigning reliability measures to each sensor node are proposed to overcome these problems and are verified experimentally. In the next chapter, fuzzy c-means clustering algorithm and minimum description length principle are employed to find the optimum number of classes existing in sonar data.

Chapter 4

DETERMINATION OF THE NUMBER OF CLASSES IN SONAR DATA

In this chapter, application of the fuzzy c-means clustering algorithm and the minimum description length principle to determine the number of classes in sonar data collected from a number of classes of targets is investigated. This chapter is organized as follows: the fuzzy c-means clustering algorithm and the minimum description length principle are introduced in Sections 4.1 and 4.2, respectively. In Section 4.3, these techniques are employed to sonar data to find the optimum number of clusters in the data.

4.1 Fuzzy c-Means Clustering Algorithm

Clustering is a tool which searches the relationships among patterns in the data set by organizing the patterns into a number of clusters, where the patterns inside a cluster show a certain degree of closeness or similarity. Cluster analysis can be divided into

two as hard and fuzzy clustering. In hard clustering, cluster boundaries are assumed to be well defined and each pattern in the data set belongs to one of the clusters with a degree of membership equal to one. However, this type of clustering cannot reflect the description of the data set when the cluster boundaries are not well defined. In such cases, fuzzy clustering is a more useful technique where each pattern in the data set is assigned to all clusters with a degree of membership μ_{ij} in [0,1]. When fuzzy clustering is used as the basis for hard clustering, pattern j is assigned to cluster k with a degree of membership equal to one if $\mu_{kj} \geq \mu_{ij}\ \forall i = 1, \cdots, c$ where $c \geq 2$ is the total number of clusters. However, it should be noted that these sets may not be disjoint when more than one maximum exists.

Fuzzy c-means clustering algorithm has been developed by Dunn [78] and extended by Bezdek [79] for the fuzzy clustering of a data set. It minimizes the following objective function with respect to fuzzy membership μ_{ij} and cluster centers $\mathbf{v_i}$.

$$J_m = \sum_{i=1}^{c} \sum_{j=1}^{N} (\mu_{ij})^m \parallel \mathbf{x_j} - \mathbf{v_i} \parallel_{\mathbf{A}}^2 \tag{4.1}$$

where

$$\parallel \mathbf{x} \parallel_{\mathbf{A}}^2 = \mathbf{x^T A x}. \tag{4.2}$$

In these equations, $\mathbf{A}$ is a $d \times d$ positive definite matrix, d is the dimension of the input patterns $\mathbf{x_j}$ $(j = 1, \cdots, N)$, N is the number of patterns, and $m > 1$ is the weighting exponent for μ_{ij} and controls the fuzziness of the resulting clusters. The fuzzy c-means clustering algorithm can be summarized as:

1) initialize the memberships μ_{ij}'s such that $\sum_{i=1}^{c} \mu_{ij} = 1$

2) compute the cluster centers $\mathbf{v_i}$'s for $i = 1, 2, \cdots, c$ using

$$\mathbf{v_i} = \frac{\sum_{j=1}^{N} (\mu_{ij})^m \mathbf{x_j}}{\sum_{j=1}^{N} (\mu_{ij})^m} \tag{4.3}$$

3) update the memberships μ_{ij}'s using

$$\mu_{ij} = \frac{\left(\frac{1}{\|\mathbf{x_j}-\mathbf{v_i}\|^2_{\mathbf{A}}}\right)^{\frac{1}{m-1}}}{\sum_{i=1}^{c}\left(\frac{1}{\|\mathbf{x_j}-\mathbf{v_i}\|^2_{\mathbf{A}}}\right)^{\frac{1}{m-1}}} \tag{4.4}$$

4) repeat second and third steps until the value of J_m no longer decreases.

The fuzzy c-partition of the data set consists of a set of fuzzy membership values μ_{ij} which can be conveniently arrayed as a $c \times N$ matrix $U = [\mu_{ij}]$. The major drawback of this algorithm is that the total number of clusters c cannot be defined *a priori*. In order to find the optimal number of clusters existing in the data set, a cluster validity criterion should be applied. A fuzzy validity criterion for fuzzy clustering algorithms has been proposed in [80]. This validity criterion depends on the data set, geometric distance measure, distance between cluster centers, and the fuzzy membership values computed by any fuzzy clustering algorithm used. This proposed fuzzy validity criterion has the following functional definition

$$S = \frac{\sum_{i=1}^{c}\sum_{j=1}^{N}\mu_{ij}^2 \parallel \mathbf{v_i} - \mathbf{x_j} \parallel^2}{N \min_{i,j} \parallel \mathbf{v_i} - \mathbf{v_j} \parallel^2} \tag{4.5}$$

Note that $\| \cdot \|$ is the usual Euclidean norm. Mathematical and numerical justification of this criterion is discussed in [80]. This criterion is a measure of the compactness and separation of the clusters. In this equation, the term $\pi \triangleq \frac{\sum_{i=1}^{c}\sum_{j=1}^{N}\mu_{ij}^2\|\mathbf{v_i}-\mathbf{x_j}\|^2}{N}$ is defined as the *compactness* of the fuzzy c-partition of the data set which is the ratio of *total variation* of the data set with respect to the fuzzy c-partition to the total number of patterns in the data set. A smaller π corresponds to a fuzzy c-partition with more compact clusters. The term $s \triangleq \min_{i,j} \parallel \mathbf{v_i} - \mathbf{v_j} \parallel^2$ is defined as the *separation* of a fuzzy c-partition where a larger s indicates larger separation between the clusters. Since $S = \pi/s$, a smaller S indicates a partition in which all the clusters are compact, and separate from each other.

4.2 Minimum Description Length Principle

The minimum description length (MDL) principle has been proposed by Rissanen as a general criterion for model selection [81]. MDL has its roots in information theory and can be considered as an estimation technique used in information theory and statistics [82, 83]. It is used to select the 'best model' from a given possible set of models which results in the minimum description length of the signal in bits. MDL criterion has two terms, the first of which is called the *data term* and corresponds to the number of bits required to represent a signal (or data) with a given model. The second term, called the *penalty term*, represents the complexity of that model. As the model complexity increases, the error in representing the data usually decreases. MDL searches a model among alternative models which trade off between model complexity and accurate data representation. MDL principle has been applied to a number of areas such as image processing [84–86], model-based multisensor fusion [87], reduction of decision graphs [88], and finding the optimal number of hidden-layer neurons in neural networks [89, 90].

MDL principle is closely related to maximum likelihood estimation (MLE). In fact, MLE is a special case of the MDL principle. Let $p(\mathbf{x}|\mathbf{\Theta})$ be a parametric class of probability functions where $\mathbf{\Theta} = (\theta_1, \cdots \theta_k)^T$ is a parameter vector and $\mathbf{x} = (x_1, \cdots, x_N)^T$ is an observation vector. In MLE, the problem is to find a $\mathbf{\Theta}^*$ which maximizes $p(\mathbf{x}|\mathbf{\Theta})$ or equivalently minimizes $\mathcal{L}(\mathbf{x}|\mathbf{\Theta}) \stackrel{\Delta}{=} -\log_2 p(\mathbf{x}|\mathbf{\Theta})$ which is called the *self-information* in information theory. $\mathcal{L}(\mathbf{x}|\mathbf{\Theta})$ represents the data term in MDL which is the *ideal code length* in bits required to code an observation $\mathbf{x}$ using model $\mathbf{\Theta}$ [83]. The penalty term $\mathcal{L}(\mathbf{\Theta})$ is the code length in bits which is required to code the model parameters. Rissanen has derived the description length of integer and real-valued parameters in [91].

The total description length required to describe data $\mathbf{x}$ using model $\mathbf{\Theta}$ is

$$\mathcal{L}(\mathbf{x}, \mathbf{\Theta}) = \mathcal{L}(\mathbf{x}|\mathbf{\Theta}) + \mathcal{L}(\mathbf{\Theta}) \tag{4.6}$$

In the MDL principle, the problem is to find a $\mathbf{\Theta}^*$ which minimizes the total description length $\mathcal{L}(\mathbf{x}, \mathbf{\Theta})$ to give the *minimum description length* $\mathcal{L}(\mathbf{x}, \mathbf{\Theta}^*)$. The main disadvantage of the MDL principle is that there is no analytical solution.

4.3 Determination of the Number of Clusters in Sonar Data

The targets employed in this study are: cylinders with radii 2.5, 5.0 and 7.5 cm, a planar target, a corner, an edge of $\theta_e = 90°$, and an acute corner of $\theta_c = 60°$. Amplitude and TOF patterns of these targets are collected with the sensing node described in Section 2.5 (the only difference is in the motor step size which is 1.8° instead of 0.9°) at 25 different locations (r, θ) for each target, from $\theta = -20°$ to $\theta = 20°$ in 10° increments, and from $r = 35$ cm to $r = 55$ cm in 5 cm increments (Figure 4.1). The target type located at range r and azimuth θ is scanned by the sensing node for scan angle $-52° \leq \alpha \leq 52°$ with 1.8° increments. The reason for using a wider range for the scan angle is the possibility that a target may still generate returns outside of the range of θ. Referring back to Figure 2.4, the angle α is always measured with respect to $\theta = 0°$ regardless of target location (r, θ). (That is, $\theta = 0°$ and $\alpha = 0°$ coincide.)

For the given scan range and motor step size, 58 ($= \frac{2 \times 52°}{1.8°}$) angular samples of each of amplitude and TOF patterns $[A_{aa}(\alpha),\ A_{bb}(\alpha),\ A_{ab}(\alpha),\ A_{ba}(\alpha)\ ;\ t_{aa}(\alpha),\ t_{bb}(\alpha),\ t_{ab}(\alpha),\ t_{ba}(\alpha)]$ are acquired at each target location. Four similar sets of scans are collected for each target type at each location, resulting in 700 (= 4 data sets × 25 locations × 7 target types) signals. This set of 700 data is referred as *training set* throughout this thesis.

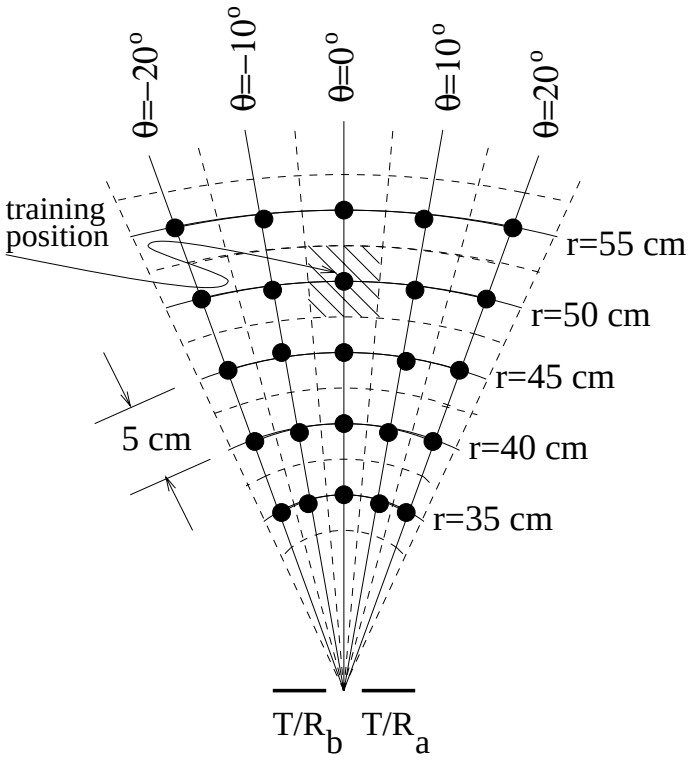


Figure 4.1: Discrete target locations.

Three different data sets are used, each one consisting of different vector representations. In each data set, seven classes (one for each target type) and 100 vector representations for each class exist. The different vector representations are:

$$\mathbf{X_I} : [A_{aa}(\alpha),\ A_{bb}(\alpha),\ \tfrac{A_{ab}(\alpha)+A_{ba}(\alpha)}{2},\ t_{aa}(\alpha),\ t_{bb}(\alpha),\ \tfrac{t_{ab}(\alpha)+t_{ba}(\alpha)}{2}]^T$$

$$\mathbf{X_{II}} : [A_{aa}(\alpha) - A_{ab}(\alpha),\ A_{bb}(\alpha) - A_{ba}(\alpha),\ t_{aa}(\alpha) - t_{ab}(\alpha),\ t_{bb}(\alpha) - t_{ba}(\alpha)]^T$$

$$\mathbf{X_{III}} : [[A_{aa}(\alpha) - A_{ab}(\alpha)][A_{bb}(\alpha) - A_{ba}(\alpha)],\ [A_{aa}(\alpha) - A_{ab}(\alpha)] + [A_{bb}(\alpha) - A_{ba}(\alpha)],$$
$$[t_{aa}(\alpha) - t_{ab}(\alpha)][t_{bb}(\alpha) - t_{ba}(\alpha)],\ [t_{aa}(\alpha) - t_{ab}(\alpha)] + [t_{bb}(\alpha) - t_{ba}(\alpha)]]^T$$

The first vector representation $\mathbf{X_I}$ is taken as the original form of the patterns without any processing, except for averaging the cross terms [$A_{ab}(\alpha)$ is averaged with $A_{ba}(\alpha)$, and $t_{ab}(\alpha)$ is averaged with $t_{ba}(\alpha)$. Since these cross terms should ideally be equal, their averages are more representative.]. The choice of the second vector representation $\mathbf{X_{II}}$ has been motivated by the target differentiation algorithm developed by Ayrulu and Barshan [21] and used with artificial neural network classifiers in [24]. The third vector

representation $\mathbf{X_{III}}$ is motivated by the differential terms which are used to assign belief values to the target types for decision fusion with Dempster-Shafer evidential reasoning and majority voting [21].

These three data sets are clustered by using the fuzzy c-means clustering algorithm with $2 \leq c \leq 10$. Here, $d = 348$ for the data set with the vector representation $\mathbf{X_I}$, $d = 232$ for the data sets with the vector representations $\mathbf{X_{II}}$ and $\mathbf{X_{III}}$, and $N = 700$ for all data sets. The matrix $\mathbf{A}$ has been chosen as the $d \times d$ identity matrix which leads to the definition of Euclidean distance resulting in spherical clusters, and m is taken as 1.3 for all data sets.

c	vector representations		
	$\mathbf{X_I}$	$\mathbf{X_{II}}$	$\mathbf{X_{III}}$
2	0.191	0.218	0.228
3	0.168	0.198	0.208
4	0.163	0.194	0.204
5	0.161	0.190	0.194
6	0.158	0.188	0.198
7	0.157	0.179	0.189
8	0.163	0.198	0.208
9	0.176	0.199	0.209
10	0.185	0.209	0.219

Table 4.1: Values of the validity function S for three data sets with vector representations $\mathbf{X_I}$, $\mathbf{X_{II}}$, and $\mathbf{X_{III}}$.

The values of the validity function S defined in Equation (4.5) have been calculated for each data set. These values are tabulated in Table 4.1. Referring to this table, minimum values of the validity function S are 0.157, 0.179 and 0.189 obtained for $c = 7$ in the data sets corresponding to the vector representations $\mathbf{X_I}$, $\mathbf{X_{II}}$ and $\mathbf{X_{III}}$, respectively. Therefore, the most suitable number of clusters for these three data sets is seven as expected. When $c = 2$, planes, edges, and cylinders with all three radii are included in one cluster and corners and acute corners are classified into another cluster for all

three data sets. When $c = 3$, planes are separated from edges and cylinders into a new cluster. Next, edges are classified into a new cluster when an extra cluster is added to the fuzzy c-means clustering algorithm (i.e., $c = 4$). Acute corners are distinguished from corners when c is further incremented by one. When $c = 6$, cylinders with $r_c = 2.5$ cm are moved to a new cluster. Finally, cylinders with $r_c = 5.0$ cm are separated from cylinders with $r_c = 7.5$ cm into the newly added cluster when $c = 7$. After this point, targets are unnecessarily divided into two or three clusters. For example, planes located along the line-of-sight of the transducer can be included in one cluster and the remaining planes are collected into another cluster, or the planes that are along the line-of-sight of the transducer can be placed in one cluster, and the planes to the left and right of the line-of-sight of the transducer are placed in two different clusters.

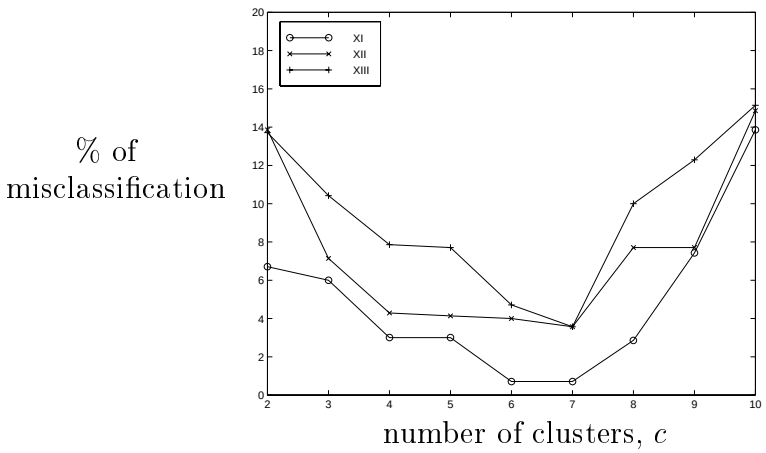


Figure 4.2: Percentage of misclassification versus the number of clusters used in the fuzzy c-means clustering algorithm for the three data sets with vector representations $\mathbf{X_I}$, $\mathbf{X_{II}}$, and $\mathbf{X_{III}}$.

The percentages of misclassification of the targets have been calculated for the same range of c. These percentages are plotted for each of the three data sets in Figure 4.2. Referring to this figure, minimum percentage of misclassification is 3.57% when $c = 7$ for the data sets containing the vector representations $\mathbf{X_{II}}$ and $\mathbf{X_{III}}$. For the data set containing the vector representation $\mathbf{X_I}$, this number is 0.71% for both $c = 6$ and $c = 7$.

Although the percentage of misclassification is equal for both $c = 6$ and $c = 7$, the cylinders with radii $r_c = 5.0$ and $r_c = 7.5$ cm are grouped into the same cluster when $c = 6$, as mentioned above. The data set consisting of the vector representation $\mathbf{X_I}$ results in the lowest percentages of misclassification compared to the other data sets. Moreover, among all three data sets, highest percentages of misclassification are obtained for the data set consisting of the vector representation $\mathbf{X_{III}}$.

4.3.1 MDL Principle for Finding the Optimal Number of Clusters in Fuzzy c-Means Clustering Algorithm

In this section, the MDL principle is employed to find the optimal number of clusters in the fuzzy c-means clustering algorithm. The model parameter is the total number of clusters c used in the algorithm. We need to find the optimal number of clusters c^* which results in the minimum description length. For any observation $\mathbf{x_j}$, we need to find $p(\mathbf{x_j}|\text{cluster } i)$ for $1 \leq i \leq c$. Using Bayes' theorem:

$$p(\mathbf{x_j}|\text{cluster } i) = \frac{p(\text{cluster } i|\mathbf{x_j})p(\mathbf{x_j})}{p(\text{cluster } i)}. \tag{4.7}$$

Here, $p(\text{cluster } i|\mathbf{x_j}) = \mu_{ij}$, $p(\mathbf{x_j}) = \frac{1}{N}$, and $p(\text{cluster } i) = \frac{1}{c}$. Then

$$p(\mathbf{x_j}|\text{cluster } i) = \frac{\mu_{ij}c}{N}. \tag{4.8}$$

and the number of bits required to code observation $\mathbf{x_j}$ in cluster i is

$$\mathcal{L}(\mathbf{x_j}|\text{cluster } i) = -\log_2 p(\mathbf{x_j}|\text{cluster } i) \tag{4.9}$$

Therefore, the number of bits required to code observation $\mathbf{x_j}$ with c clusters is

$$\mathcal{L}(\mathbf{x_j}|c) = -\sum_{i=1}^{c} \log_2 p(\mathbf{x_j}|\text{cluster } i) \tag{4.10}$$

Finally, the data term in the total description length can be expressed as

$$\mathcal{L}(\mathbf{x}|c) = -\frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{c} \log_2 \left(\frac{\mu_{ij}c}{N}\right) \tag{4.11}$$

which is the average number of bits required to describe pattern $\mathbf{x}$. For the penalty term, we need to find the number of bits required to code the model parameters. For a model having c clusters, d-dimensional cluster centers $\mathbf{v_i}$'s for $i = 1, \cdots, c$ must be coded. According to reference [91], the description length of c real-valued parameters of dimension d is $\frac{1}{2}c \log_2 d$. Therefore, the penalty term in the total description length can be expressed as

$$\mathcal{L}(c) = \frac{1}{2}c \log_2 d \tag{4.12}$$

and the total description length is

$$\mathcal{L}(\mathbf{x}, c) = -\frac{1}{N} \sum_{j=1}^{N} \sum_{i=1}^{c} \log_2 \left(\frac{\mu_{ij} c}{N} \right) + \frac{1}{2} c \log_2 d. \tag{4.13}$$

The total description length given in Equation (4.13) is calculated when the total number of clusters c varies between 2 and 10 for all three vector representations $\mathbf{X_I}$, $\mathbf{X_{II}}$, and $\mathbf{X_{III}}$. The data term, the penalty term, and the total description length for these vector representations are plotted in Figure 4.3(a)-(c) respectively. Referring to these figures, it can be observed that both the data term and the penalty term increase with increasing c. In fact, to represent an observation $\mathbf{x_j}$, c membership values μ_{ij} for $1 \leq i \leq c$ must be coded. With increasing c values, the number of membership values to be coded increases, hence the data term increases. However, after $c = 7$ the average number of bits required to code a membership value is almost constant around 20, 18, 22 bits for the vector representations $\mathbf{X_I}$, $\mathbf{X_{II}}$, and $\mathbf{X_{III}}$ respectively [Figure 4.3(d)]. With this raw interpretation of the MDL principle, minimum values of the total description length of Equation (4.13) are obtained at $c^* = 2$ for all three vector representations.

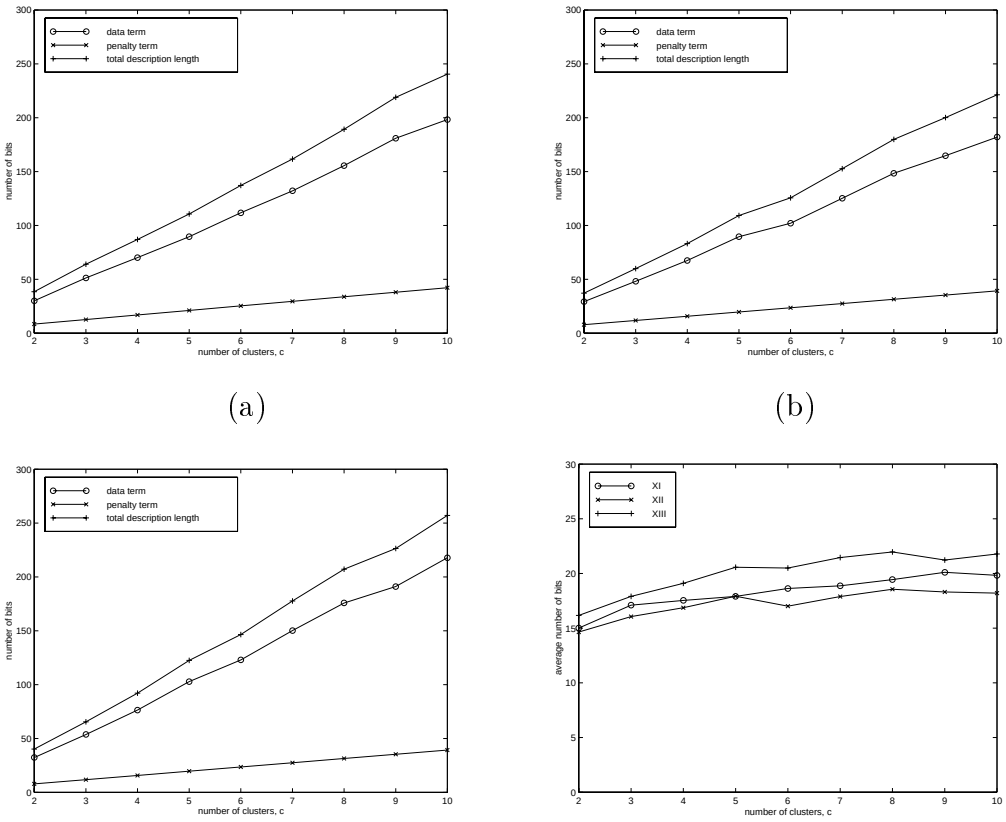


Figure 4.3: The data term, penalty term, and the total description length versus the total number of clusters c in the fuzzy c-means clustering algorithm for the vector representation (a) $\mathbf{X_I}$, (b) $\mathbf{X_{II}}$, and (c) $\mathbf{X_{III}}$. (d) The average number of bits required to code a membership value for all three vector representations.

However, we know at least that there exists more than two clusters in our problem. Some other objective functions related to the MDL principle may be more convenient to our problem instead of raw interpretation of the MDL principle. These kinds of objective functions are also used in a number of problems such as feature discrimination [92], model selection in image segmentation [86], and finding the optimal number of hidden-layer neurons in neural networks [89]. In our problem, a relatively compact and separate fuzzy c-partition giving the minimum classification error with minimum possible number of clusters is preferable. Therefore, the objective function given below can be employed to find the most suitable number of clusters c in the fuzzy c-means clustering algorithm:

$$O(c) = K_1 S_c + K_2 p_c(E) + K_3 \mathcal{L}(c) \tag{4.14}$$

Here, S_c is the validity criterion used in the previous section which is related to the compactness and separation of a fuzzy c-partition, $p_c(E)$ is the probability of misclassification which is equal to $\frac{1}{100} \times$ percentage of misclassification, $\mathcal{L}(c)$ is the penalty term used in total description length in the MDL principle which is equal to the cost of having c clusters, and K_1, K_2, and K_3 are the weighting factors that adjust the contribution of these three terms which need to be determined. Then, the problem is to find the c^* value which minimizes the objective function above.

Suitable K_1, K_2, and K_3 values are determined by considering some limiting cases. In the first limiting case, assuming that adding one more cluster does not bring any improvement in compactness and separation of the clusters (i.e., $S_{c+1} - S_c = 0$), then the cost of adding one more cluster should be compensated by a decrease in the probability of misclassification. That is,

$$K_2[p_{c+1}(E) - p_c(E)] + K_3[\mathcal{L}(c+1) - \mathcal{L}(c)] < 0 \tag{4.15}$$

In the second limiting case, assume that adding one more cluster does not bring any improvement in the probability of misclassification (i.e., $p_{c+1}(E) - p_c(E) = 0$). This time, the cost of adding one more cluster should be compensated by a decrease in S which means more compact and separate fuzzy c-partitioning. That is,

$$K_1[S_{c+1} - S_c] + K_3[\mathcal{L}(c+1) - \mathcal{L}(c)] < 0 \tag{4.16}$$

The relative magnitudes of K_1 and K_2 with respect to each other is application dependent. If the probability of misclassification is more important than the compactness and separation of the partition, then take $K_1 < K_2$ and vice versa.

To find the K_1, K_2, and K_3 values, K_3 is set to 1 and K_1 and K_2 are found by using the two limiting cases which are expressed in Inequalities (4.15) and (4.16) for all three vector representations. In these inequalities, the expected values of $S_{c+1} - S_c$ and $p_{c+1}(E) - p_c(E)$ are used. From Figure 4.2, the expected value of the decrease in the percentage of misclassification is around 2% and from Table 4.1, the expected value of the

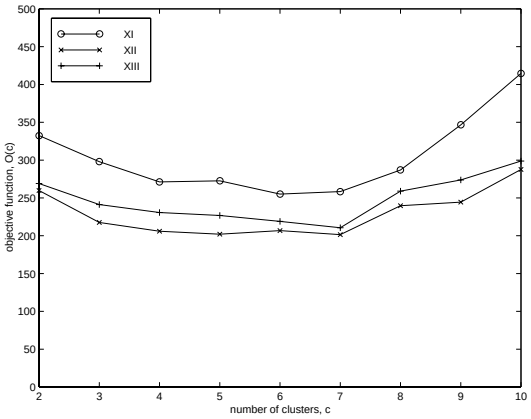


Figure 4.4: The values of the objective function, $O(c)$, versus the total number of clusters c in the fuzzy c-means clustering algorithm for the vector representations $\mathbf{X_I}$, $\mathbf{X_{II}}$, and $\mathbf{X_{III}}$.

decrease in S values is around 0.010 for all three vector representations. Boundary values for K_1 are 630, 393, and 393 for the corresponding vector representations. Similarly, boundary values for K_2 are 352, 191, and 194. In this study, K_1 and K_2 values are taken as two times as large as the boundary values. For all three vector representations, the values of the objective function $O(c)$ for $2 \leq c \leq 10$ are plotted in Figure 4.4. Referring to this figure, minimum values of the objective function for the second and the third vector representations are obtained for $c = 7$. For the first vector representation, minimum value of the objective function is obtained for $c = 6$.

In this chapter, the optimum number of classes existing in the training set is found by employing both the fuzzy c-means clustering algorithm and the minimum description length principle. In the next chapter, various representations of input sonar signals, two different training algorithms, and different network structures are employed to improve the performance of neural network classifiers in target classification and localization.

Chapter 5

NEURAL NETWORKS FOR IMPROVED TARGET DIFFERENTIATION AND LOCALIZATION

Although it is theoretically possible to differentiate all target types included in this study, due to the vulnerability of the echo amplitude to noise and the deficiency of the physical models used to model the reflections from each target type in accounting for second or higher-order reflections in the environment, only three target types can be differentiated by the differentiation algorithm. Decision fusion through majority voting and Dempster-Shafer evidential reasoning, described in Chapter 2, and voting with preference ordering and reliability measures in Chapter 3 bring substantial improvement in the classification. Moreover, the neural network classifier employed in [24] is able to differentiate all target types, making more effective use of the available data. The neural network's performance indicates that the original training set does contain sufficient information to differentiate

all target types, but the differentiation algorithm is not able to resolve this identifying information in real sonar systems. Although trained on a discrete and relatively coarse grid, the network is able to interpolate between the grid locations and offers higher resolution (especially in azimuth) than that implied by the grid size. These results recommend wider use of neural networks as robust pattern classifiers in the area of sensor-based robotics, in particular, for target differentiation and localization. In this chapter, improvement of the performance of neural network classifier is considered by employing various representations of input sonar signals, different training algorithms, and different network structures.

This chapter is organized as follows: multi-layer feed-forward neural networks and modular neural networks are briefly reviewed in Section 5.1 and 5.2, respectively. Two training algorithms, namely back-propagation and generating-shrinking algorithms, are described in Section 5.3. In Section 5.4, preprocessing techniques employed prior to neural network classifiers are briefly described. In Section 5.5, various types of input signals to the neural network classifiers are proposed. In Section 5.6, the effect of these input signals and training algorithms on the performance of neural networks in target classification and localization are investigated experimentally.

5.1 Multi-layer Feed-Forward Neural Networks

Multi-layer feed-forward neural networks (multi-layer perceptrons) have been widely used in areas such as target detection and classification [93], speech processing [94], system identification [95], control theory [96], medical applications [97], and character recognition [98]. They consist of an input layer, one or more hidden layers, and a single output layer, each comprised of a number of units called *neurons*. These networks have three distinctive characteristics: The model of each neuron includes a smooth nonlinearity, the network contains one or more hidden layers to extract progressively

more meaningful features, and the network exhibits a high degree of connectivity. Due to the presence of distributed form of nonlinearity and high degree of connectivity, theoretical analysis of multi-layer perceptrons is difficult. These networks are trained to compute the boundaries of decision regions in the form of connection weights and biases by using training algorithms. In this study, two training algorithms are employed, namely, back-propagation and generating-shrinking algorithms which are briefly reviewed in the next section.

Two well-known methods for determining the number of hidden-layer neurons in feed-forward neural networks are *pruning* and *enlarging* [99]. Pruning begins with a relatively large number of hidden-layer neurons and eliminates unused neurons according to some criterion. Enlarging begins with relatively small number of hidden-layer neurons and gradually increases their number until learning occurs.

It is proven that the multi-layer perceptron approximates the Bayes optimal discriminant function in the mean-square sense when it is trained as a classifier using the back-propagation algorithm with infinitely many training samples and uniform losses [100]. The outputs of this classifier also represent the corresponding posterior probabilities [100]. However, the accuracy of the approximation is limited by the architecture of the network being trained such that if the hidden-layer neurons are too few, then the approximation will not provide a good match. Fortunately, it is not dependent on the number of layers and the type of activation function (nonlinearity) used.

5.2 Modular Neural Networks

Modular neural networks are proposed by several researchers in the light of the studies of human and animal brains suggesting the existence of considerable specialization in different parts of the brain [99,101]. A network is said to be modular if the task performed

by the network can be decomposed into two or more modules which uses distinct input without communicating with each other and the outputs of these modules are combined with an integrating unit which decides how the outputs of these modules should be combined to form the output of the whole network. These networks offer fast training times since the modules which are used to construct the modular neural network are structurally simpler than the non-modular neural network which is designed to perform the same task. Each module can be trained independently and in parallel, and the sum of the number of iterations needed to train each module is less than the sum of the number of iterations required for training a non-modular neural network. Moreover, modular neural networks are expected to have better generalization capability since each module performs a simpler task than the corresponding non-modular neural network. The training times and performances of modular and non-modular back-propagation neural networks are compared for multi-class problems in speech recognition and character recognition areas in [101].

5.3 Training Algorithms

5.3.1 The Back-Propagation Algorithm

The back-propagation algorithm is used frequently due to its simplicity, extraction power of useful information from examples, and capacity of implicit information storage in the form of connection weights, and applicability to binary or real-valued patterns [102]. While training with the back-propagation algorithm, a set of *training patterns* is represented to the network and propagated forward to determine the resulting signal at the output. Back-propagation algorithm is a gradient-descent procedure that minimizes the error at the output. The average error at a particular cycle of the back-propagation algorithm is the average of the Euclidean distance between the actual output of the

network and the desired output for all training patterns:

$$E_{\text{ave}} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}||d_i - o_i||^2 \tag{5.1}$$

Here N is the number of training patterns ($N = 700$), d_i is the desired output for the ith pattern and o_i is the actual output of the network for the ith pattern. The nonlinearity used in this network is a sigmoid function of the form

$$\varphi(v) = \frac{1}{1 + e^{-v}} \tag{5.2}$$

where v is the input to a neuron. The error is back-propagated through the network in order to adjust the connection weights and biases. Adjustment of these quantities is proportional to the descent gradient of sum of squared errors with a constant called the *learning rate* chosen between zero and one. The speed of the training procedure is very slow with too small learning rates, but there can be stability problems if the learning rate is chosen too large. To avoid these problems, a second term in the adjustment equation, called the *momentum term*, is added [103]. This term is proportional to the previous adjustment through a *momentum constant*. In this study, the stopping criterion we have used while training networks with the back-propagation algorithm is as follows: The training is stopped either when the average error is reduced to 0.001 or if a maximum of 10000 epochs is reached, whichever occurs earlier. The second case occurs very rarely.

5.3.2 Generating-Shrinking Algorithm

The generating-shrinking algorithm first builds and then shrinks or prunes a four-layer feed-forward neural network, offering fast convergence rate and 100% correct classification on the training set as reported in [104] on scale-invariant texture discrimination. The network used in the same study consists of two hidden layers with equal numbers of neurons which is initially set equal to the number of training patterns. Pre-determined connection weights are assigned. Then, the hidden layers

are pruned while preserving 100% correct classification rate on the training set. The algorithm is based on the assumption that only one output neuron can take the value one (the winning neuron) and the remaining output neurons take the value zero. Since the initial connection weights take deterministic values, the network has analytically known generalization behavior. At the input layer, a pre-fixed reference number n_r that can take values between zero and infinity is used as an additional input to control the generalization capability of the network. The algorithm achieves scale-invariant generalization behavior as n_r approaches zero, and behaves like a nearest-neighborhood classifier as it tends to infinity. A comparison with the back-propagation algorithm in [104] indicates that the generating-shrinking algorithm does not have the convergence problems of the back-propagation algorithm and has a substantially faster convergence rate (2.2 s versus 1260 s) and perfect generalization capability (100% versus 68%), although both networks have 100% correct classification rate on the training set. For further details of this algorithm, the reader can refer to [104].

5.4 Preprocessing of the Input Signals

In this section, we give a brief description of the preprocessing techniques used on the input signals to the neural networks considered in this study.

5.4.1 Ordinary Fourier Transform

Fourier analysis is a well known and widely-used technique in signal processing to study the spectral behavior of a signal [105]. The discrete Fourier transform (DFT) of a signal $f(n)$ is defined as:

$$F(k) = \mathcal{F}\{f(n)\} \triangleq \frac{1}{N} \sum_{n=0}^{N-1} f(n) e^{-\frac{2\pi}{N} jnk} \tag{5.3}$$

where N is the length of the signal $f(n)$.

5.4.2 Fractional Fourier Transform

The ath-order fractional Fourier transform is a generalization of the ordinary Fourier transform such that the first-order fractional Fourier transform is the ordinary Fourier transform and the zeroth-order fractional Fourier transform corresponds to the function itself [106,107]. The transform has been studied extensively since the early nineties with a view to applications in wave propagation, optics and optical signal processing [108–111], time- and space-frequency analysis [112, 113], pattern recognition [114], digital signal [115–118] and image processing [119–121], and other areas. Most applications are based on replacing the ordinary Fourier transform with the fractional Fourier transform. Since the latter has an additional degree of freedom (the order parameter a), it is often possible to generalize and improve upon previous results.

The ath-order fractional Fourier transform $f_a(u)$ of the function $f(u)$ is defined for $0 < |a| < 2$ as [117, 122]

$$\begin{aligned} f_a(u) &\triangleq \int_{-\infty}^{\infty} K_a(u,u') f(u') du' \\ K_a(u,u') &\triangleq A_\phi \exp\left[j\pi(u^2\cot\phi - 2uu'\csc\phi + u'^2\cot\phi)\right] \\ \text{where} \quad A_\phi &= \frac{\exp[-j(\pi\text{sgn}(\phi)/4 - \phi/2)]}{|\sin\phi|^{1/2}} \quad \text{and} \quad \phi = \frac{a\pi}{2} \end{aligned} \tag{5.4}$$

The kernel $K_a(u,u')$ approaches $\delta(u-u')$ and $\delta(u+u')$ as a approaches 0 and ± 2, respectively, and are defined as such at these values. The fractional Fourier transform reduces to the ordinary Fourier transform when $a = 1$. The transform is linear and index additive; that is, the a_1th-order fractional Fourier transform of the a_2th-order fractional Fourier transform of a function is equal to the $(a_1 + a_2)$th-order fractional Fourier transform. An important property of the fractional Fourier transform relating it to time-frequency (or space-frequency) concepts is its close relationship to the Wigner distribution [123]. The ath-order fractional Fourier transform of a function corresponds to a rotation of the Wigner distribution of the function by an angle $a\pi/2$ in the time-frequency plane. Moreover, digital implementation of the fractional Fourier transform

is as efficient as that of the ordinary Fourier transform in the sense that it can also be computed in the order of $N \log N$ time, where N is the number of sample points or the signal length [124].

With a similar notation as in the case of DFT, the ath-order discrete fractional Fourier transform (DFRT) of $\mathbf{f}$, denoted $\mathbf{f_a}$, can be expressed as

$$\mathbf{f_a} = \mathbf{F^a f} \tag{5.5}$$

where $\mathbf{F^a}$ is the $N \times N$ DFRT matrix which corresponds to the ath power of the ordinary DFT matrix $\mathbf{F}$. However, it should be noted that there are certain subtleties and ambiguities in defining the power function, for which we refer the reader to [125, 126].

The DFRT can be used to approximately compute the continuous fractional Fourier transform. That is, it can be used to approximately map the samples of the original function into the samples of its fractional Fourier transform. As with the ordinary DFT, the value of N should be chosen at least as large as the time- or space-bandwidth product of the signals in question.

5.4.3 Hartley Transform

Hartley transform [127] is another widely-used technique in signal processing in areas such as image compression [128] and adaptive filtering [129]. The discrete Hartley transform (DHT) of a signal $f(n)$ is defined as:

$$H(k) = \mathcal{H}\{f(n)\} \triangleq \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} f(n) \, \mathrm{cas}\left(\frac{2\pi}{N} nk\right) \tag{5.6}$$

where $\mathrm{cas}(x) \triangleq \cos(x) + \sin(x)$ and N is the length of the signal $f(n)$. There is a close relationship between the Fourier transform and Hartley transform such that if the DFT of a signal $f(n)$ is expressed as $F(k) = F_R(k) - jF_I(k)$, then its DHT is related to the real and imaginary parts of the DFT by $H(k) = F_R(k) + F_I(k)$. The DHT can also be

represented in matrix notation as

$$\mathbf{h_1} = \mathbf{Hf} \tag{5.7}$$

where $\mathbf{f}$ is an $N \times 1$ column vector, $\mathbf{H}$ is the $N \times N$ DHT matrix, and $\mathbf{h_1}$ is the DHT of $\mathbf{f}$.

5.4.4 Wavelet Transform

Wavelet transform is a relatively new analytical tool for engineers, scientists and mathematicians for time-frequency analysis, and a new basis for representing functions [130]. The discrete wavelet transform (DWT) of a function $f(t) \in \mathbf{L}^2$ can be obtained from the expansion:

$$f(t) = \sum_{k=-\infty}^{\infty} c(k)\varphi_k(t) + \sum_{j=0}^{\infty}\sum_{k=-\infty}^{\infty} d(j,k)\psi_{j,k}(t) \tag{5.8}$$

where

$$c(k) = < f(t), \varphi_k(t) >= \int f(t)\varphi_k(t)dt \tag{5.9}$$

$$d(j,k) = < f(t), \psi_{j,k}(t) >= \int f(t)\psi_{j,k}(t)dt \tag{5.10}$$

The coefficients $\{c(k)\}_{k=-\infty}^{\infty}$ and $\{d(j,k)\}_{j=0,k=-\infty}^{\infty,\infty}$ are called DWT of the function $f(t)$. These coefficients completely describe the original signal and can be used in a way similar to Fourier series coefficients. At this point, it is necessary to consider the functions $\varphi_k(t)$ and $\psi_{j,k}(t)$ in Equation (5.8). A set of scaling functions in terms of integer translations of a basic scaling function $\varphi(t)$ is represented as $\varphi_k(t) = \varphi(t-k);\ k \in \mathbf{Z}$ and $V_\circ = \text{Span}\{\varphi_k(t)\} \subset \mathbf{L}^2$. A family of functions generated from the basic scaling function $\varphi(t)$ by scaling and translation is represented by $\varphi_{j,k}(t) = 2^{j/2}\varphi(2^j t - k)$ and $V_j = \text{Span}\{\varphi_{j,k}(t)\}$ such that $\cdots \subset V_\circ \subset V_1 \subset V_2 \subset \cdots \subset \mathbf{L}^2,\ V_{-\infty} = 0,\ V_\infty = \mathbf{L}^2$.

Since $\varphi(t) \in V_1$, it can be represented in terms of basis functions of V_1. Then:

$$\varphi(t) = \sum_{n=0}^{M-1} h(n)\varphi(2t-n) \tag{5.11}$$

where $h(n)$, $n = 0, \cdots, M-1$ is called the *scaling filter*. Important features of the signal can be better described by not using $\varphi_{j,k}(t)$ with increasing j to increase the size of the subspace spanned by the scaling functions but by defining a slightly different set of functions that spans the differences between spaces spanned by various scales of $\varphi(t)$. These functions are called *wavelet functions*. If the orthogonal complement of V_j in V_{j+1} is denoted as W_j, then

$$\begin{aligned} V_1 &= V_\circ \oplus W_\circ \\ V_2 &= V_\circ \oplus W_\circ \oplus W_1 \\ &\vdots \\ \mathbf{L}^2 &= V_\circ \oplus W_\circ \oplus W_1 \oplus \cdots \end{aligned} \tag{5.12}$$

where $\oplus$ is orthogonal sum operator.

Since these wavelets reside in the space spanned by the next narrower scaling function, they can be represented in terms of the scaling function as:

$$\psi(t) = \sum_{n=0}^{M-1} g(n)\varphi(2t-n) \tag{5.13}$$

where $g(n)$ is called *wavelet filter* simply related to the scaling filter by

$$g(n) = (-1)^n h(M-n-1) \quad n = 0, \cdots, M-1 \tag{5.14}$$

and M is the length of $h(n)$.

Finally, the procedure of finding the wavelet transform coefficients can be summarized as:

$$c_j(k) = \sum_{m=0}^{M-1} h(m-2k)c_{j+1}(m) \tag{5.15}$$

$$d_j(k) = \sum_{m=0}^{M-1} g(m-2k)c_{j+1}(m) \tag{5.16}$$

Here, $k = 0, 1, \cdots, 2^j N - 1$ where N is the number of samples of the original signal that should be a power of 2. This equation shows that the scaling and wavelet coefficients

at different scales j can be obtained by convolving scaling coefficients at scale $j+1$ by $h(-n)$ and $g(-n)$ and then downsampling (take every other term) (Figure 5.1(a)).

In the reconstruction part,

$$c_{j+1}(k) = 2\left[\sum_{m=0}^{M-1} c_j(m)h(k-2m) + \sum_{m=0}^{M-1} d_j(m)g(k-2m)\right] \quad k = 0, 1, \cdots, 2^{j+1}N - 1 \quad (5.17)$$

This equation shows that $c_{j+1}(k)$'s can be evaluated by upsampling the scaling and wavelet coefficients, which means doubling their length by inserting zeros between each term, then convolving them with $h(n)$ and $g(n)$ respectively, and finally adding the resulting terms and multiplying by two (Figure 5.1(b)). Usually, $c_\circ(k)$'s are taken as the samples of the original signal.

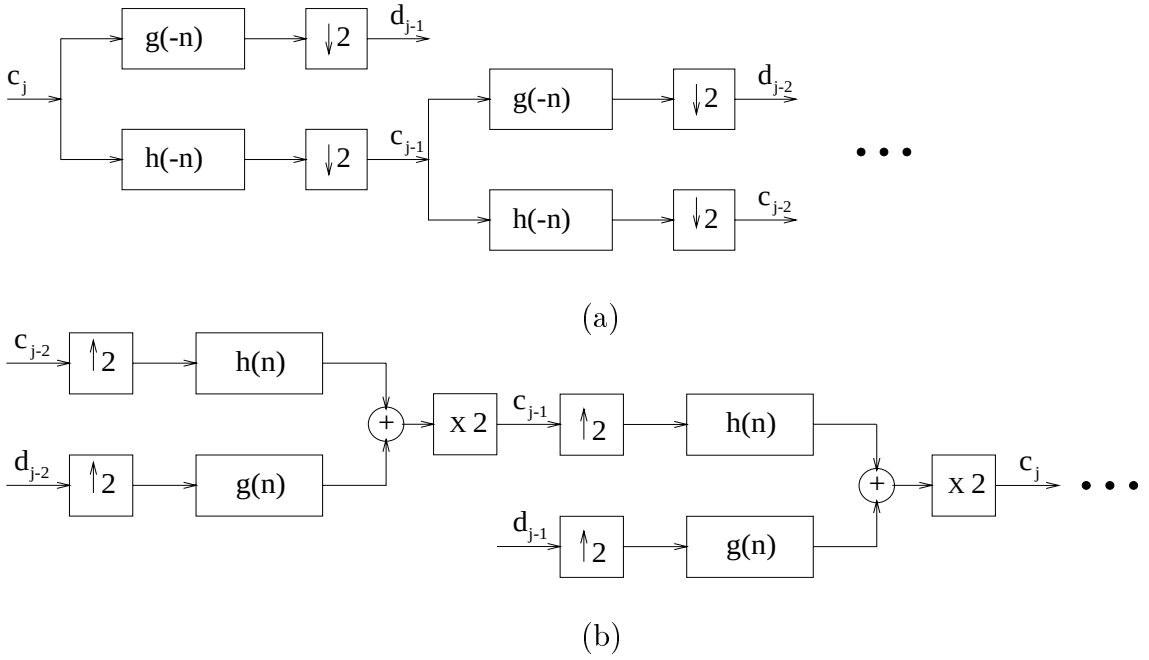


Figure 5.1: (a) Analysis and (b) synthesis of DWT coefficients.

5.4.5 Self-Organizing Feature Map

Self-organizing neural networks are generated by unsupervised learning algorithms that have the ability to form internal representation of the network that model the underlying structure of the input data. These networks are commonly used to solve the scaling problem encountered in supervised learning procedures. However, it is not recommended to use them by themselves for pattern classification or other decision-making processes [99]. Instead, best results are achieved with these networks when they are used as feature extractors prior to a linear classifier or a supervised learning process for pattern classification. The most commonly used algorithm for generating self-organizing neural networks is Kohonen's self-organizing feature-mapping (KSOFM) algorithm [131]. In this algorithm, weights are adjusted from the input layer towards the output layer where the output neurons are interconnected with local connections. These output neurons are geometrically organized in one, two, three, or even higher dimensions. This algorithm can be summarized as follows:

- initialize the weights randomly
- present new input from the training set
- find the winning neuron at the output layer
- select the neighborhood of this output neuron
- update weights from input towards selected output neurons
- continue with the second step until no considerable changes in the weights occur

For further details of this algorithm, the reader can refer to [99].

5.5 Input Signals to the Neural Network

An important issue in target differentiation with neural networks is to select those input signals to the network that carry sufficient information to differentiate all target types. Input signals resulting in a minimal network configuration (in terms of the number of layers and the number of neurons in these layers) with minimum classification error are preferable. There are many different ways of choosing input signals to the network. Apart from the sonar signals themselves, differential amplitude and TOF patterns have been used frequently in previous studies on sonar sensing [6, 14, 20, 21, 24]. In this thesis, amplitude and TOF patterns and their differentials are used either in their raw form or after some preprocessing as inputs to the neural networks.

We considered the samples of the following 30 different signals as alternative inputs to the neural networks:

$I_1 : A_{aa}(\alpha),\ A_{bb}(\alpha),\ \frac{A_{ab}(\alpha)+A_{ba}(\alpha)}{2},\ t_{aa}(\alpha),\ t_{bb}(\alpha),$ and $\frac{t_{ab}(\alpha)+t_{ba}(\alpha)}{2}$

$I_2 : A_{aa}(\alpha) - A_{ab}(\alpha),\ A_{bb}(\alpha) - A_{ba}(\alpha),\ t_{aa}(\alpha) - t_{ab}(\alpha),$ and $t_{bb}(\alpha) - t_{ba}(\alpha)$

$I_3 : [A_{aa}(\alpha) - A_{ab}(\alpha)][A_{bb}(\alpha) - A_{ba}(\alpha)],\ \ [A_{aa}(\alpha) - A_{ab}(\alpha)] + [A_{bb}(\alpha) - A_{ba}(\alpha)],$
$[t_{aa}(\alpha) - t_{ab}(\alpha)][t_{bb}(\alpha) - t_{ba}(\alpha)],$ and $\ [t_{aa}(\alpha) - t_{ab}(\alpha)] + [t_{bb}(\alpha) - t_{ba}(\alpha)]$

$I_4 - I_{12}$: discrete Fourier transform $[\mathcal{F}(I_i),\ i = 1, 2, 3]$

$I_{13} - I_{15}$: discrete fractional Fourier transform of I_1, I_2, I_3 at different orders a
$[\mathcal{F}^a(I_i),\ i = 1, 2, 3]$

$I_{16} - I_{18}$: discrete Hartley transform $[\mathcal{H}(I_i),\ i = 1, 2, 3]$

$I_{19} - I_{27}$: discrete wavelet transform of I_1, I_2, I_3 at different resolutions

$I_{28} - I_{30}$: features extracted by using KSOFM [KSOFM$(I_i),\ i = 1, 2, 3$]

To the best of our knowledge, these input signals have not been used earlier for target classification with sonar. The first signal I_1 is taken as the original form of the patterns without any processing, except for averaging the cross terms [$A_{ab}(\alpha)$ is averaged with $A_{ba}(\alpha)$, and $t_{ab}(\alpha)$ is averaged with $t_{ba}(\alpha)$. Since these cross terms should ideally

be equal, their averages are more representative.]. The choice of the second signal I_2 has been motivated by the target differentiation algorithm developed by Ayrulu and Barshan [21] and used with neural network classifiers in [24]. The third input signal I_3 is motivated by the differential terms which are used to assign belief values to the target types classified by the target differentiation algorithm [21]. These three input signals have been used both in their raw form and after taking their discrete ordinary and fractional Fourier, Hartley, and wavelet transforms, as well as after feature extraction by KSOFM. DWTs of each signal at different resolution levels j are used. Initially, DWT of each signal at resolution level $j = -1$ is used as the input [DWT(I_i), $i = 1, 2, 3$]. Secondly, only the low-frequency component of the DWT, c_{-1}s, are employed [LFC(DWT(I_i))$_1$]. Finally, the low-frequency component of DWT at resolution $j = -2$, c_{-2}s, are used [LFC(DWT(I_i))$_2$]. The low-frequency components of the DWT are more similar to the original signal. When the resolution is further decreased, the performance of the network deteriorates since the number of samples in the low-frequency component decreases with decreasing resolution level j. For this reason, we have stopped at resolution $j = -2$. While obtaining these DWTs, original signal samples are taken as $c_\circ$, and the scaling filter whose first twelve coefficients are given in Table 5.1 [132] is used. Note that this filter is symmetrical with respect to $n = 0$. The low-frequency component of DWT at resolution $j = -2$ corresponds to the frequency domain information between 0 and $\frac{\pi}{4}$ rad. In order to make a fair comparison between the discrete Fourier transform and LFC(DWT(I_i))$_2$, the low-frequency component of DFT [LFC($\mathcal{F}(I_i)$)] corresponding to the same frequency interval as the frequency content of LFC(DWT(I_i))$_2$ is considered. In this case, the magnitude of the low-frequency component of DFT, |LFC($\mathcal{F}(I_i)$)|, is also employed. The ath-order discrete fractional Fourier transforms of these three input signal representations, for a values varying from 0.05 to 0.95 with 0.05 increments, are also used. Finally, the features extracted by using KSOFM are used as input signals [KSOFM(I_i), $i = 1, 2, 3$]. In this case, the extracted features are used both prior to neural networks trained by the two training algorithms and prior to linear classifiers

designed by using a least-squares approach.

n	0	1	2	3	4	5	6	7	8	9	10	11
$h(n)$	0.542	0.307	−0.035	−0.078	0.023	−0.030	0.012	−0.013	0.006	0.006	−0.003	−0.002

Table 5.1: First 12 coefficients of the scaling filter $h(n)$ which is symmetrical with respect to $n = 0$.

5.6 Experimental Studies

The aim of this study is to employ neural networks to identify and resolve parameter relations embedded in the characteristics of sonar echo returns from *all* target types considered, in a robust and compact manner in real time. Performance of neural network classifiers is affected by the choice of parameters related to the network structure, training algorithm, and input signals, as well as parameter initialization [38]. In this work, various input signal representations described in the previous section and two different training algorithms, reviewed in Section 5.3, are considered to improve the performance of neural networks in target classification and localization with sonar.

The same ultrasonic sensor node used in all of our previous studies is used with step size 1.8° (Figure 2.10). The same training set is used for training which was generated by scanning the targets: cylinders with radii 2.5, 5.0 and 7.5 cm, a planar target, a corner, an edge of $\theta_e = 90°$, and an acute corner of $\theta_c = 60°$ with the sensor node at 25 sensing locations of Figure 4.1. Neural networks trained with the back-propagation algorithm consist of one input, one hidden, and one output layer. The number of input-layer neurons is determined by the total number of samples of the amplitude and TOF patterns used by a particular type of input signal, described in Section 5.5. These numbers for the networks trained with the back-propagation algorithm are tabulated in Tables 5.2 and 5.3. For example, for the input signal I_1, the original forms of the amplitude and TOF patterns are used without any processing, except for averaging the

cross terms as explained in the previous section. After averaging, there are six patterns each with 58 samples; therefore 348 ($= 6 \times 58$) input units are used. For the second input signal I_2, four amplitude and TOF differentials are used, therefore 232 ($= 4 \times 58$) input units are needed. Similarly, for the input signal I_3, there are also four input patterns and 232 is the number of input neurons. When the DHT of I_1, I_2, and I_3 is taken, the resulting signal has the same number of samples as the original signal. Since the real and imaginary parts of the DFT and DFRTs of I_1, I_2, and I_3 are downsampled by 2, the number of input-layer neurons needed for the DFT and DFRTs of I_1, I_2, and I_3 is the same as that needed for the corresponding original signals. In the case of DFT, using only the low-frequency components, the number of input samples is reduced by $\frac{1}{4}$. For the DWT, the number of samples used needs to be a power of two. Therefore, the number of samples (58) is increased to 64 by padding with zeroes. In this case, for DWT(I_1), we have $6 \times 64 = 384$, for DWT(I_2) and DWT(I_3), we have $4 \times 64 = 256$ input units to the neural network. For the KSOFM algorithm, a two-dimensional output layer (7×25) is used which is presented as input to the neural network. Therefore 175 ($= 7 \times 25$) input-layer neurons are needed. The number of hidden-layer neurons is determined by *enlarging*. The number of output-layer neurons is 21. The first seven neurons encode the target type. The next seven represent the target range r which is binary coded with a resolution of 0.25 cm. The last seven neurons represent the azimuth θ of the target with respect to the line-of-sight of the sensing unit, which is binary coded with resolution 0.5°.

In addition, modular network structures for each type of input signal have been implemented in which three separate networks for target type, range, and azimuth, each trained with the back-propagation algorithm, are employed. The different network structures implemented in this study are illustrated in Figure 5.2 for the input signal I_2. In the modular case, each of the three modules has the same number of input-layer neurons as the corresponding non-modular network. The number of hidden-layer neurons

input signal	input	hidden	output
I_1	348	120	21
I_2	232	100	21
I_3	232	105	21
$\mathcal{F}(I_1)$	348	65	21
LFC($\mathcal{F}(I_1)$)	96	55	21
$\lvert$LFC($\mathcal{F}(I_1)$)$\rvert$	48	73	21
$\mathcal{F}(I_2)$	232	35	21
LFC($\mathcal{F}(I_2)$)	64	35	21
$\lvert$LFC($\mathcal{F}(I_2)$)$\rvert$	32	43	21
$\mathcal{F}(I_3)$	232	53	21
LFC($\mathcal{F}(I_3)$)	64	45	21
$\lvert$LFC($\mathcal{F}(I_3)$)$\rvert$	32	60	21
$\mathcal{F}^a(I_1)$	348	150	21
$\mathcal{F}^a(I_2)$	232	60	21
$\mathcal{F}^a(I_3)$	232	60	21
$\mathcal{H}(I_1)$	348	53	21
$\mathcal{H}(I_2)$	232	48	21
$\mathcal{H}(I_3)$	232	35	21
DWT(I_1)	384	50	21
LFC(DWT(I_1))$_1$	192	50	21
LFC(DWT(I_1))$_2$	96	66	21
DWT(I_2)	256	50	21
LFC(DWT(I_2))$_1$	128	68	21
LFC(DWT(I_2))$_2$	64	54	21
DWT(I_3)	256	65	21
LFC(DWT(I_3))$_1$	128	64	21
LFC(DWT(I_3))$_2$	64	62	21
KSOFM(I_1)	175	55	21
KSOFM(I_2)	175	50	21
KSOFM(I_3)	175	50	21

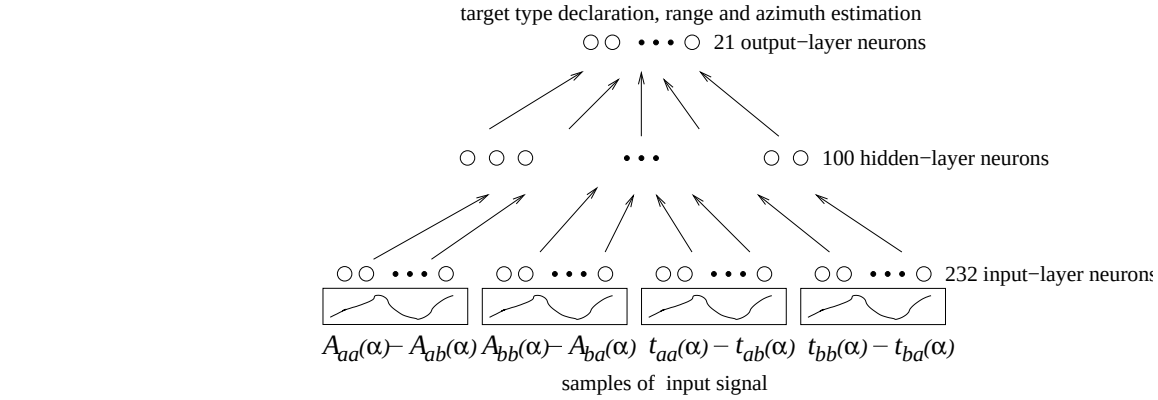
Table 5.2: Number of neurons used in the input, hidden and output layers of the non-modular networks trained with the back-propagation algorithm.

is again determined by enlarging and varies as shown in Table 5.3. The number of output-layer neurons of each module is 7. Referring to Tables 5.2 and 5.3, the maximum number of total neurons in the network layers is obtained with the input signals $\mathcal{F}^a(I_1)$ and I_1 for non-modular and modular network structures, respectively, the minimum number of

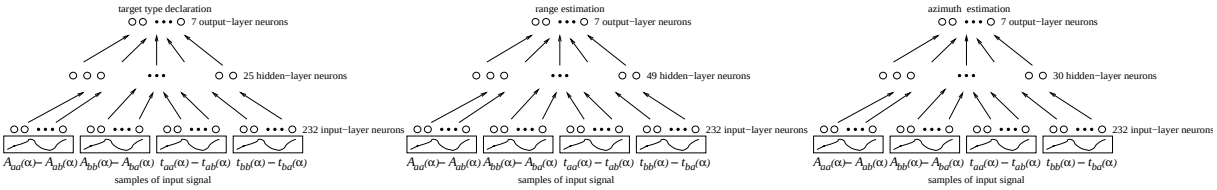
input signal	input	target type	r	θ	output
I_1	348	90	125	75	7
I_2	232	25	49	30	7
I_3	232	30	65	45	7
$\mathcal{F}(I_1)$	348	50	58	38	7
LFC($\mathcal{F}(I_1)$)	96	23	48	30	7
|LFC($\mathcal{F}(I_1)$)|	48	13	25	15	7
$\mathcal{F}(I_2)$	232	13	13	13	7
LFC($\mathcal{F}(I_2)$)	64	15	18	13	7
|LFC($\mathcal{F}(I_2)$)|	32	18	28	23	7
$\mathcal{F}(I_3)$	232	30	30	28	7
LFC($\mathcal{F}(I_3)$)	64	20	23	23	7
|LFC($\mathcal{F}(I_3)$)|	32	23	25	28	7
$\mathcal{F}^a(I_1)$	348	30	58	38	7
$\mathcal{F}^a(I_2)$	232	19	25	15	7
$\mathcal{F}^a(I_3)$	232	15	30	30	7
$\mathcal{H}(I_1)$	348	18	28	23	7
$\mathcal{H}(I_2)$	232	15	18	13	7
$\mathcal{H}(I_3)$	232	13	18	21	7
DWT(I_1)	384	30	60	45	7
LFC(DWT(I_1))$_1$	192	35	55	50	7
LFC(DWT(I_1))$_2$	96	50	95	80	7
DWT(I_2)	256	34	37	30	7
LFC(DWT(I_2))$_1$	128	26	35	35	7
LFC(DWT(I_2))$_2$	64	45	75	36	7
DWT(I_3)	256	50	50	50	7
LFC(DWT(I_3))$_1$	128	35	40	38	7
LFC(DWT(I_3))$_2$	64	58	80	50	7
KSOFM(I_1)	175	34	55	45	7
KSOFM(I_2)	175	30	50	40	7
KSOFM(I_3)	175	30	50	42	7

Table 5.3: Number of neurons used in the input, hidden and output layers of each modular network designed for target classification, r and θ estimation. Note that the number of input and output neurons of the modules are equal.

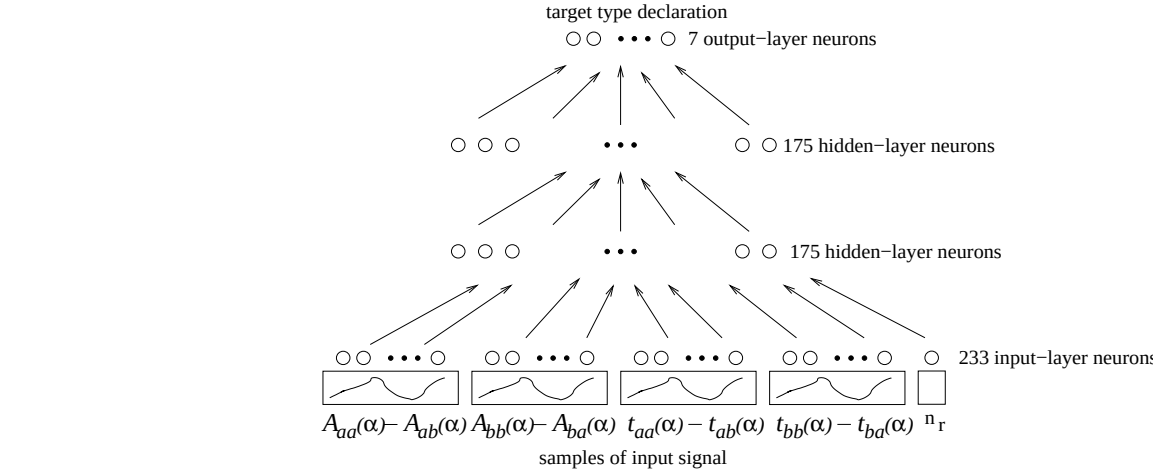
total neurons in the network layers is obtained with the input signal |LFC($\mathcal{F}(I_2)$)| for both cases.



(a)



(b)



(c)

Figure 5.2: The structure of the (a) non-modular and (b) modular networks trained with the back-propagation algorithm, (c) non-modular network trained with the generating-shrinking algorithm when the input signal I_2 is used.

Neural networks using the same input signals are also trained with the generating-shrinking algorithm. This algorithm can only be applied to target type classification since it is based on the assumption that only one output neuron takes the value one (the winning neuron) and the others are zero. For this reason, range and azimuth estimation cannot be made with this approach. In these networks, the number of input-layer neurons for each type of input signal is determined as described above for back-propagation networks, except that there is an additional input neuron for the reference number n_r. The reference number n_r is taken as 0.01 after making a number of simulations with n_r varying between 0.005 and 0.1. The output layer has seven neurons. Initially, each of the two hidden layers has 700 neurons (equal to the number of training patterns) which is reduced by one fourth to 174 or 175 after training. Since the number of neurons in the two hidden layers are approximately equal (174 or 175) and the number of output neurons is fixed for all types of input signals, the complexity of these networks can be assessed by the number of their input neurons.

In order to measure the performance of the neural networks in target classification and localization, three different test data sets are generated. Initially, each target type is placed in turn in each of the 25 training positions shown in Figure 4.1. Four sets of patterns are collected for each combination of target type and location, again resulting in 700 sets of experimentally acquired patterns. This set is referred as *test set I* throughout this thesis. The test data are not collected at the same time as the training data. Rather, each target is first moved through all the grid locations and a complete training set is fully completed (700 sets of patterns). The test data for the grid locations are obtained later by repositioning the objects at the grid locations and acquiring another 700 sets of patterns. This means that there will inevitably be some differences in the object positions and orientations, as well as the ambient conditions (i.e., temperature and humidity) even though the targets are nominally placed at the same grid points. In the testing stage, the targets are not presented to the sensing node following the same order used in training. Rather, a random strategy is followed.

Next, the targets are situated arbitrarily in the continuous estimation space and not necessarily confined to the 25 locations of Figure 4.1. This second set of test data was acquired with about a month's delay after collecting the training data which is referred as *test set II* throughout this thesis. Randomly generated locations within the area shown in Figure 4.1, not necessarily corresponding to one of the 25 grid locations, are used as target positions. The r, θ values corresponding to these locations are generated by using the uniform random number generator in MATLAB. The range for r is [32.5 cm, 57.5 cm] and that for θ is $[-25^\circ, 25^\circ]$.

Finally, we have carried out tests with targets not scanned during training which are slightly different in size, shape, or roughness than the targets used for training. These are two smooth cylinders of radii 4 cm and 10 cm, a cylinder of radius 7.5 cm covered with bubbled packing material, a 60° smooth edge, and a plane covered with bubbled packing material. The packing material with bubbles has a honeycomb pattern of uniformly distributed circular bubbles of diameter 1.0 cm and height 0.3 cm, with a center-to-center separation of 1.2 cm. The test data are collected at the 25 grid locations used for training. Note that this set of test data is referred as *test set III* throughout this thesis.

The different network structures are tested with the test sets I, II, and III. Based on these data, neural networks trained with the back-propagation algorithm estimate the target type, range, and azimuth; those trained with the generating-shrinking algorithm determine only the target type.

For non-modular and modular networks trained with the back-propagation algorithm, the resulting average percentages over all target types for correct type classification, correct range and correct azimuth estimation are given in Tables 5.4−5.6 for test sets I−III respectively. In these tables, the numbers before the parentheses are for non-modular networks, whereas the numbers in the parentheses are for modular networks trained with the back-propagation algorithm. In all of these tables, the best results

input signal	% of correct classif.	% of correct r estimation				% of correct θ estimation			
		error tolerance ϵ_r				error tolerance ϵ_θ			
		± 0.125 cm	± 1 cm	$\pm$ 5 cm	± 10 cm	$\pm 0.25^\circ$	$\pm 2^\circ$	$\pm 10^\circ$	$\pm 20^\circ$
I_1	88(88)	30(33)	41(46)	63(70)	86(87)	65(65)	76(72)	87(84)	97(97)
I_2	95(95)	74(73)	77(88)	87(93)	93(96)	89(95)	92(96)	95(97)	97(99)
I_3	86(88)	79(73)	82(75)	89(83)	94(91)	83(87)	89(91)	95(95)	97(98)
$\mathcal{F}(I_1)$	97(98)	64(72)	69(73)	86(87)	96(95)	86(94)	93(96)	96(98)	100(100)
LFC($\mathcal{F}(I_1)$)	96(97)	56(70)	64(73)	86(88)	95(97)	84(92)	90(96)	96(96)	100(99)
|LFC($\mathcal{F}(I_1)$)|	88(86)	28(45)	35(52)	68(77)	88(93)	65(55)	70(59)	86(79)	95(90)
$\mathcal{F}(I_2)$	93(89)	59(60)	64(65)	79(78)	89(90)	76(73)	81(86)	88(91)	93(96)
LFC($\mathcal{F}(I_2)$)	99(95)	63(68)	72(74)	85(86)	94(92)	91(89)	93(91)	96(96)	99(98)
|LFC($\mathcal{F}(I_2)$)|	86(95)	35(54)	42(60)	73(80)	96(94)	39(56)	50(65)	71(86)	86(95)
$\mathcal{F}(I_3)$	86(90)	54(62)	61(65)	77(77)	89(89)	70(77)	76(82)	85(88)	94(94)
LFC($\mathcal{F}(I_3)$)	91(85)	60(60)	68(65)	82(78)	92(90)	77(78)	81(83)	88(89)	96(96)
|LFC($\mathcal{F}(I_3)$)|	74(82)	34(41)	42(49)	65(72)	85(90)	30(53)	39(60)	62(78)	83(90)
$\mathcal{F}^a(I_1)$	100(96)	75(62)	79(66)	89(86)	97(96)	93(76)	96(79)	97(92)	100(99)
$\mathcal{F}^a(I_2)$	98(98)	67(68)	71(76)	83(87)	92(95)	80(86)	84(89)	90(95)	96(98)
$\mathcal{F}^a(I_3)$	90(93)	61(59)	68(62)	83(80)	92(90)	76(75)	82(79)	88(88)	95(94)
$\mathcal{H}(I_1)$	99(97)	59(54)	68(60)	85(81)	94(94)	84(84)	89(87)	95(95)	98(99)
$\mathcal{H}(I_2)$	98(97)	67(62)	72(68)	85(80)	93(90)	80(84)	85(86)	91(93)	96(99)
$\mathcal{H}(I_3)$	87(81)	59(46)	66(51)	80(69)	90(89)	73(79)	80(84)	89(90)	95(95)
DWT(I_1)	82(74)	15(21)	30(27)	59(59)	80(82)	46(51)	58(63)	77(80)	94(94)
LFC(DWT(I_1))$_1$	85(98)	18(21)	28(33)	58(59)	82(79)	54(49)	65(62)	80(79)	95(94)
LFC(DWT(I_1))$_2$	98(99)	71(80)	76(82)	87(91)	95(96)	90(92)	93(93)	97(98)	100(100)
DWT(I_2)	92(96)	63(64)	69(69)	84(82)	93(92)	85(87)	88(90)	93(94)	96(96)
LFC(DWT(I_2))$_1$	95(97)	65(66)	70(71)	84(84)	94(91)	87(88)	90(90)	94(94)	97(96)
LFC(DWT(I_2))$_2$	89(84)	28(32)	34(44)	58(68)	84(88)	58(53)	68(61)	86(80)	95(92)
DWT(I_3)	86(89)	58(58)	62(62)	76(76)	93(89)	85(76)	88(80)	93(88)	96(94)
LFC(DWT(I_3))$_1$	82(91)	56(61)	60(66)	75(78)	89(87)	73(79)	77(83)	86(89)	93(94)
LFC(DWT(I_3))$_2$	83(79)	29(33)	37(44)	63(69)	83(88)	53(41)	65(52)	78(75)	87(89)
KSOFM(I_1)	75(74)	17(14)	25(23)	49(46)	80(72)	64(61)	67(64)	81(79)	90(89)
KSOFM(I_2)	78(76)	22(19)	28(28)	59(57)	88(81)	69(66)	73(71)	86(85)	92(93)
KSOFM(I_3)	66(63)	24(21)	30(31)	57(55)	84(81)	51(49)	54(51)	78(75)	89(87)

Table 5.4: The percentages of correct classification, range (r) and azimuth (θ) estimation for test set I.

are given for the DFRT of these signals with the order $0.05 \leq a \leq 0.95$ [133]. The best results for DFRT of the signals I_1, I_2, and I_3 are obtained with $a = 0.25$, 0.55, and 0.65, respectively. A range or azimuth estimate is considered correct if it is within an error tolerance of ϵ_r or ϵ_θ of the actual range or azimuth, respectively. For test set I (Table 5.4), the highest average percentages of correct classification of 100%

and 99% obtained with the input signal $\mathcal{F}^a(I_1)$, and LFC(DWT(I_1))$_2$ for non-modular and modular networks, respectively. For non-modular networks, the highest average percentages of correct azimuth estimation are achieved with $\mathcal{F}^a(I_1)$ and lies in the range $93-96\%$ for $|\epsilon_\theta| = 0.25-2^\circ$. It is 97% for $|\epsilon_\theta| = 10^\circ$ which is obtained with the signals $\mathcal{F}^a(I_1)$ and LFC(DWT(I_1))$_2$, and 100% for $|\epsilon_\theta| = 20^\circ$ is obtained with the signals $\mathcal{F}^a(I_1)$, LFC(DWT(I_1))$_2$, and $\mathcal{F}(I_1)$. The highest average percentage of correct range estimation lies in the range $79-82\%$ and is obtained with the input signal I_3 for the error tolerances $|\epsilon_r| = 0.125-1$ cm, and it is 89% for $|\epsilon_r| = 5$ cm which is achieved with both the signals I_3 and $\mathcal{F}^a(I_1)$, and it is 97% obtained with $\mathcal{F}(I_1)$ for $|\epsilon_r| = 10$ cm. For modular networks, the highest average percentages of correct azimuth estimation is 95% is achieved with I_2 for the error tolerance level $|\epsilon_\theta| = 0.25^\circ$ and 96% for $|\epsilon_\theta| = 2^\circ$ obtained with the signals I_2 and $\mathcal{F}(I_1)$. For the error tolerance levels $|\epsilon_\theta| = 10^\circ$ and 20°, the highest average percentages of correct azimuth estimation are 98 and 100% which are obtained with both $\mathcal{F}(I_1)$ and LFC(DWT(I_1))$_2$. The highest average percentage of correct range estimation for $|\epsilon_r| = 0.125$ cm is 80% is obtained with LFC(DWT(I_1))$_2$. The highest average percentage of correct range estimation are 88 and 93% for $|\epsilon_r| = 1$ and 5 cm obtained with the input signal I_2. For $|\epsilon_r| = 10$ cm, it is 96% which is achieved with the input signals I_2, $\mathcal{F}(I_1)$, and LFC(DWT(I_1))$_2$. For both modular and non-modular cases, when wavelet transformed signals are used, the results are comparable to the results of the original signal. However, employing only the low-frequency component of the wavelet transform at the resolution level $j = -1$ (i.e., c_{-1}) results in better classification and estimation performance than employing both c_{-1} and d_{-1}. While classification and estimation performance further increases by using the low-frequency component of the wavelet transform at the resolution level $j = -2$ for the input signal I_1, it decreases for I_2 and I_3. For both non-modular and modular network structures, employing only the low-frequency component of the Fourier transform gives better classification and estimation performance than employing all frequency components of the Fourier transformed signals for the input signals I_2 and I_3. These results are also better than the results obtained

with the low-frequency component of the wavelet transform at the resolution level $j = -2$ for these input signals. For the input signal I_1, comparable classification and estimation performance is obtained with employing only the low-frequency component of the Fourier transform and with employing all frequency components of the Fourier transform. Still the results obtained with the low-frequency component of the wavelet transform at the resolution level $j = -2$ for the input signal I_1 is better than the results obtained with the low-frequency component of the Fourier transform of this input signal. However, employing only the magnitude of the low-frequency components of the Fourier transform even worsens the results obtained with employing all frequency components of the Fourier transformed signals for all input signal representations.

For test set II (Table 5.5), the maximum correct target classification percentages of 100% (non-modular network structure) and 99% (modular structure) are maintained when the input signals $\mathcal{F}^a(I_1)$ and LFC(DWT$(I_1)_2$ are used, respectively. These values are the same as those achieved at the grid positions. As expected, the percentages for the non-grid test positions can be lower than those for the grid test positions by 0 to 30 percentage points; the networks give the best results when a test target is situated exactly at one of the training sites. Noting that the networks are trained only at 25 locations and at grid spacings of 5 cm and 10°, it can be concluded from the percentage of correct range and azimuth estimates obtained at error tolerances of $|\epsilon_r| =$ 0.125 cm and 1 cm and $|\epsilon_\theta| = 0.25°$ and 2°, that the networks demonstrate the ability to interpolate between the training grid locations. Thus, the neural network maintains a certain spatial continuity between its input and output and does not haphazardly map positions which are not drawn from the 25 locations of Figure 4.1. The correct target type percentages in the corresponding tables are quite high and the accuracy of the range/azimuth estimates would be acceptable for most of the input signals in many applications. If better estimates are required, this can be achieved by reducing the training grid spacing in Figure 4.1. Moreover, these percentages for the modular network structures are slightly better than those for neural networks in which type classification

input signal	% of correct classif.	% of correct r estimation				% of correct θ estimation			
		error tolerance ϵ_r				error tolerance ϵ_θ			
		± 0.125 cm	± 1 cm	$\pm$ 5 cm	± 10 cm	$\pm 0.25^\circ$	$\pm 2^\circ$	$\pm 10^\circ$	$\pm 20^\circ$
I_1	88(88)	17(18)	32(30)	55(56)	78(83)	37(38)	47(47)	75(74)	91(94)
I_2	90(93)	59(60)	63(69)	78(83)	88(88)	70(71)	75(76)	92(97)	94(98)
I_3	58(59)	63(60)	63(62)	76(76)	83(85)	66(69)	74(73)	93(93)	94(97)
$\mathcal{F}(I_1)$	96(98)	53(57)	54(57)	81(75)	91(88)	69(72)	77(77)	89(98)	98(98)
LFC($\mathcal{F}(I_1)$)	96(97)	52(59)	58(62)	82(83)	89(89)	69(69)	75(74)	83(83)	98(98)
|LFC($\mathcal{F}(I_1)$)|	86(82)	20(37)	28(45)	64(72)	86(88)	57(53)	66(59)	78(74)	88(88)
$\mathcal{F}(I_2)$	89(92)	52(51)	53(52)	67(68)	80(80)	60(59)	65(68)	81(92)	83(95)
LFC($\mathcal{F}(I_2)$)	98(95)	54(56)	57(58)	74(70)	83(80)	69(69)	72(73)	95(90)	97(92)
|LFC($\mathcal{F}(I_2)$)|	83(90)	21(42)	31(50)	64(74)	86(92)	39(51)	49(60)	71(75)	81(87)
$\mathcal{F}(I_3)$	84(87)	48(51)	52(53)	65(68)	77(80)	57(60)	63(65)	82(84)	89(86)
LFC($\mathcal{F}(I_3)$)	90(85)	56(53)	56(54)	74(73)	85(85)	61(62)	65(67)	87(86)	91(91)
|LFC($\mathcal{F}(I_3)$)|	74(81)	25(36)	34(43)	57(60)	80(86)	30(48)	39(56)	62(78)	81(87)
$\mathcal{F}^a(I_1)$	100(96)	59(53)	60(55)	79(79)	89(88)	70(63)	75(68)	97(97)	100(99)
$\mathcal{F}^a(I_2)$	92(92)	55(56)	55(59)	67(71)	78(83)	62(65)	67(68)	85(91)	90(92)
$\mathcal{F}^a(I_3)$	83(85)	53(52)	53(54)	72(71)	81(79)	61(60)	70(65)	85(80)	89(88)
$\mathcal{H}(I_1)$	92(96)	52(51)	55(54)	76(77)	87(89)	68(67)	74(73)	93(95)	96(99)
$\mathcal{H}(I_2)$	93(95)	55(52)	58(52)	71(68)	83(82)	62(66)	68(71)	86(94)	90(96)
$\mathcal{H}(I_3)$	77(79)	50(44)	51(45)	72(66)	83(83)	60(61)	65(68)	81(86)	87(87)
DWT(I_1)	82(74)	12(14)	24(20)	50(53)	76(79)	26(29)	37(38)	64(64)	87(89)
LFC(DWT(I_1))$_1$	85(98)	11(13)	22(22)	50(53)	75(75)	33(31)	41(43)	70(71)	87(91)
LFC(DWT(I_1))$_2$	98(99)	60(64)	60(64)	76(79)	91(89)	71(72)	77(77)	96(94)	96(95)
DWT(I_2)	92(93)	53(54)	53(57)	72(71)	85(81)	65(66)	67(69)	87(90)	92(92)
LFC(DWT(I_2))$_1$	91(94)	53(56)	53(56)	70(71)	80(80)	68(66)	72(70)	91(88)	91(90)
LFC(DWT(I_2))$_2$	86(80)	16(20)	28(29)	51(60)	80(79)	33(28)	40(34)	74(72)	86(88)
DWT(I_3)	82(85)	49(51)	53(52)	68(67)	78(81)	57(59)	63(65)	85(85)	87(88)
LFC(DWT(I_3))$_1$	80(86)	52(54)	52(54)	68(65)	80(77)	60(62)	67(68)	85(86)	88(90)
LFC(DWT(I_3))$_2$	80(78)	21(20)	30(32)	60(62)	81(83)	28(23)	38(31)	65(66)	84(84)
KSOFM(I_1)	75(73)	12(10)	19(18)	45(41)	77(69)	38(34)	40(37)	75(69)	88(86)
KSOFM(I_2)	78(76)	19(16)	23(21)	53(52)	82(78)	39(38)	45(42)	77(76)	88(87)
KSOFM(I_3)	65(61)	21(19)	26(25)	51(51)	78(73)	29(27)	34(33)	69(67)	81(80)

Table 5.5: The percentages of correct classification, range (r) and azimuth (θ) estimation for test set II.

and range and azimuth estimation are done simultaneously.

For test set III (Table 5.6), a maximum correct target classification percentage of 89% for both non-modular and modular network structures is obtained when the input signals $\mathcal{H}(I_1)$ (non-modular network structure) and $\mathcal{F}^a(I_1)$ (modular structure) are used, respectively. When the non-modular network trained with the back-propagation

input signal	% of correct classif.	% of correct r estimation				% of correct θ estimation			
		error tolerance ϵ_r				error tolerance ϵ_θ			
		± 0.125 cm	± 1 cm	$\pm$ 5 cm	± 10 cm	$\pm 0.25^\circ$	$\pm 2^\circ$	$\pm 10^\circ$	$\pm 20^\circ$
I_1	85(73)	18(21)	28(32)	49(55)	74(76)	35(40)	45(45)	61(56)	80(72)
I_2	78(80)	59(60)	59(65)	72(77)	83(84)	68(70)	73(75)	75(76)	76(80)
I_3	57(54)	60(59)	60(59)	69(69)	80(80)	64(68)	72(75)	73(78)	74(79)
$\mathcal{F}(I_1)$	77(74)	56(58)	57(58)	73(76)	87(83)	68(69)	78(77)	81(77)	81(77)
LFC($\mathcal{F}(I_1)$)	77(78)	52(59)	56(59)	73(73)	82(85)	69(68)	75(74)	83(83)	85(85)
|LFC($\mathcal{F}(I_1)$)|	68(68)	19(31)	25(36)	57(62)	81(82)	55(50)	60(56)	73(66)	78(74)
$\mathcal{F}(I_2)$	79(76)	50(53)	52(54)	59(67)	76(80)	54(61)	62(70)	71(77)	76(82)
LFC($\mathcal{F}(I_2)$)	84(81)	54(56)	57(58)	73(70)	83(79)	69(68)	72(73)	85(80)	87(86)
|LFC($\mathcal{F}(I_2)$)|	63(70)	21(35)	30(41)	60(67)	84(88)	34(42)	40(52)	68(65)	80(83)
$\mathcal{F}(I_3)$	74(76)	47(52)	48(52)	63(62)	76(75)	62(63)	69(72)	78(76)	81(79)
LFC($\mathcal{F}(I_3)$)	77(74)	52(53)	55(54)	68(66)	79(76)	61(62)	65(67)	71(73)	85(87)
|LFC($\mathcal{F}(I_3)$)|	65(70)	23(30)	31(38)	57(60)	79(82)	29(40)	36(48)	53(72)	72(79)
$\mathcal{F}^a(I_1)$	83(89)	61(55)	63(55)	77(72)	90(82)	67(67)	71(70)	71(80)	71(83)
$\mathcal{F}^a(I_2)$	81(79)	55(56)	56(57)	68(70)	79(79)	64(65)	70(72)	71(73)	73(77)
$\mathcal{F}^a(I_3)$	77(79)	52(53)	53(53)	65(64)	76(72)	62(62)	69(67)	73(75)	77(80)
$\mathcal{H}(I_1)$	89(87)	53(51)	54(52)	71(70)	79(80)	69(72)	76(76)	80(83)	80(83)
$\mathcal{H}(I_2)$	80(81)	56(52)	58(53)	72(66)	85(79)	65(66)	70(69)	74(74)	75(81)
$\mathcal{H}(I_3)$	75(72)	51(45)	53(45)	66(60)	78(78)	62(64)	69(70)	73(75)	73(76)
DWT(I_1)	78(75)	12(15)	23(19)	50(51)	75(80)	27(32)	35(45)	53(62)	79(80)
LFC(DWT(I_1))$_1$	69(84)	12(14)	18(27)	47(50)	78(70)	32(33)	41(40)	59(60)	80(77)
LFC(DWT(I_1))$_2$	85(83)	56(63)	58(63)	68(74)	82(85)	67(71)	69(75)	76(80)	76(80)
DWT(I_2)	82(80)	53(54)	54(55)	71(68)	85(83)	65(69)	69(73)	77(75)	79(73)
LFC(DWT(I_2))$_1$	80(84)	53(55)	56(56)	69(68)	79(79)	67(68)	71(72)	78(73)	78(73)
LFC(DWT(I_2))$_2$	76(74)	16(19)	22(28)	48(51)	75(74)	32(33)	39(39)	59(57)	73(73)
DWT(I_3)	73(75)	49(50)	49(50)	59(63)	75(75)	67(61)	72(67)	76(76)	79(78)
LFC(DWT(I_3))$_1$	74(80)	50(52)	50(52)	60(63)	73(75)	62(63)	68(69)	73(72)	76(74)
LFC(DWT(I_3))$_2$	73(72)	17(20)	26(30)	51(52)	73(75)	30(23)	40(32)	56(52)	72(68)
KSOFM(I_1)	73(72)	9(7)	13(12)	35(33)	60(56)	32(31)	34(32)	51(50)	65(65)
KSOFM(I_2)	75(74)	17(15)	21(21)	56(55)	85(81)	44(43)	47(46)	67(66)	76(76)
KSOFM(I_3)	66(64)	16(14)	19(20)	47(46)	73(71)	32(31)	36(35)	60(59)	83(82)

Table 5.6: Average percentages of correct classification, range (r) and azimuth (θ) estimation for test set III.

algorithm is tested with test set III, there is a decrease of 11.7 percentage points on the average of all the different input signals compared to the testing results for test set I. This number is 6.7 percentage points for the modular network trained with the back-propagation algorithm. For both non-modular and modular network structures, comparable average percentages of correct range and azimuth estimation are obtained

for both test set I and II. Overall, we can conclude that the networks exhibit some degree of robustness to variations in target shape, size and roughness. Again, overall performances of the modular network structures are slightly better than those of non-modular structures.

input signal	% of correct classif.	% of correct r estimation				% of correct θ estimation			
		error tolerance ϵ_r				error tolerance ϵ_θ			
		$\pm$0.125 cm	$\pm$1 cm	$\pm$ 5 cm	$\pm$10 cm	$\pm 0.25^o$	$\pm 2^o$	$\pm 10^o$	$\pm 20^o$
KSOFM(I_1)	81(81)[78]	33(21)[20]	37(27)[23]	61(55)[50]	85(79)[74]	75(65)[46]	76(68)[46]	88(88)[68]	94(91)[77]
KSOFM(I_2)	85(85)[77]	41(26)[28]	44(30)[30]	71(59)[58]	90(84)[80]	80(65)[47]	82(68)[48]	93(88)[63]	97(88)[76]
KSOFM(I_3)	73(73)[67]	42(28)[28]	45(34)[30]	69(60)[59]	86(78)[81]	64(58)[44]	67(63)[46]	85(81)[69]	94(84)84

Table 5.7: Average percentages of correct classification, range (r) and azimuth (θ) estimation for KSOFM used prior to a linear classifier. The numbers before the parentheses are for test set I, the numbers in the parentheses are for test set II, whereas the numbers in the brackets are for test set III.

The results obtained with KSOFM used prior to linear classifiers are given in Table 5.7. This combination results in better classification performance than when the KSOFM is applied prior to neural networks, the results of which are given in the last three rows of Tables 5.4–5.6. The classification and azimuth estimation performance of a linear classifier using features extracted by KSOFM are also comparable to those obtained with the corresponding unprocessed signals. However, range estimation performance deteriorates dramatically compared to the results obtained with the corresponding unprocessed signals (Table 5.7).

For networks trained with the generating-shrinking algorithm, the resulting average percentages over all target types for correct type classification are given in Table 5.8. Referring to this table, the maximum average percentage of correct classification is 98% for both test data I and II and is obtained with the input signal representation LFC($\mathcal{F}(I_1)$) and is followed by LFC(DWT(I_1))$_1$, LFC(DWT(I_1))$_2$, $\mathcal{F}(I_1)$, $\mathcal{H}(I_1)$, and $\mathcal{F}^a(I_1)$. It is 92% for test data III which is obtained with the input signal representation $\mathcal{F}(I_1)$ and is followed by $\mathcal{F}^a(I_1)$, and $\mathcal{H}(I_1)$. In this case, resulting percentages are almost comparable for all input signal representations except the features obtained by

input signal	Test set		
	I	II	III
I_1	95	95	89
I_2	90	90	78
I_3	76	76	68
$\mathcal{F}(I_1)$	97	97	92
LFC($\mathcal{F}(I_1)$)	98	98	86
|LFC($\mathcal{F}(I_1)$)|	97	97	84
$\mathcal{F}(I_2)$	95	95	82
LFC($\mathcal{F}(I_2)$)	96	96	81
|LFC($\mathcal{F}(I_2)$)|	94	94	75
$\mathcal{F}(I_3)$	83	83	69
LFC($\mathcal{F}(I_3)$)	88	88	75
|LFC($\mathcal{F}(I_3)$)|	83	83	71
$\mathcal{F}^a(I_1)$	97	97	91
$\mathcal{F}^a(I_2)$	96	96	83
$\mathcal{F}^a(I_3)$	84	83	71
$\mathcal{H}(I_1)$	97	97	91
$\mathcal{H}(I_2)$	95	95	81
$\mathcal{H}(I_3)$	83	83	71
DWT(I_1)	95	95	89
LFC(DWT(I_1))$_1$	97	97	89
LFC(DWT(I_1))$_2$	97	97	88
DWT(I_2)	91	91	80
LFC(DWT(I_2))$_1$	90	90	79
LFC(DWT(I_2))$_2$	90	90	79
DWT(I_3)	75	75	67
LFC(DWT(I_3))$_1$	77	77	68
LFC(DWT(I_3))$_2$	80	80	71
KSOFM(I_1)	5	8	5
KSOFM(I_2)	13	11	9
KSOFM(I_3)	8	5	6

Table 5.8: The percentages of correct classification for networks trained with the generating-shrinking algorithm for the three test sets.

using KSOFM which are much lower ($\leq 13\%$). When arbitrary test positions are used (test data II), compared to the results of test data I the decreases in the percentages of the networks trained by employing the generating-shrinking algorithm are much smaller than those of the modular and non-modular structures trained by employing

the back-propagation algorithm. Unlike the latter case, for most of the input signal representations, the two results are identical. In a few cases, there is $\pm 1 - 3\%$ difference.

To illustrate the distribution of average correct classification and localization percentages to the target types, the percentages of correct classification and correct range and azimuth estimation for each target type are provided in Tables 5.9 and 5.10 for the non-modular network trained with the back-propagation algorithm when the input signal I_2 is used. In most cases, the same trend exhibited in these tables is maintained for various input signals by the different network structures.

target type	% of correct classif.	% of correct r estimation				% of correct θ estimation			
		error tolerance ϵ_r				error tolerance ϵ_θ			
		± 0.125 cm	± 1 cm	$\pm$ 5 cm	± 10 cm	$\pm 0.25°$	$\pm 2°$	$\pm 10°$	$\pm 20°$
plane	100(90)	62(47)	66(50)	78(69)	88(80)	91(66)	96(68)	98(93)	99(92)
corner	99(100)	90(72)	90(72)	92(86)	95(91)	88(81)	90(85)	92(90)	93(90)
edge ($\theta_e = 90°$)	99(96)	51(49)	59(56)	82(78)	92(90)	72(60)	77(69)	89(81)	97(95)
acute corner ($\theta_c = 60°$)	98(99)	81(71)	83(75)	91(83)	95(88)	85(79)	87(80)	93(90)	97(92)
cylinder ($r_c = 2.5$ cm)	90(88)	77(55)	80(60)	89(78)	94(87)	97(79)	97(83)	97(95)	98(99)
cylinder ($r_c = 5.0$ cm)	89(70)	75(61)	77(65)	82(78)	89(91)	98(73)	98(78)	99(98)	99(96)
cylinder ($r_c = 7.5$ cm)	92(86)	82(55)	86(60)	92(76)	98(86)	95(54)	97(60)	99(94)	99(96)
average	95(90)	74(59)	77(63)	87(78)	93(88)	89(70)	92(75)	95(92)	97(94)

Table 5.9: The percentages of correct classification, range (r) and azimuth (θ) estimation for the non-modular network trained with the back-propagation algorithm when the input signal I_2 is used. The numbers before the parentheses are for test set I, whereas the numbers given in parentheses are for test set II.

Statistics over ten non-modular networks trained with the back-propagation algorithm using different initial conditions for the connection weights are provided in Table 5.11 for the input signal I_2. In this table, the numbers before the parentheses are the means of the average percentages of correct classification, range (r) and azimuth (θ) estimation, whereas the numbers given in parentheses are the standard deviations of the average percentages of correct classification, range (r) and azimuth (θ) estimation.

In this chapter, various input signal representations, two different training algorithms, and three different network structures are considered for neural networks to improve their

target type	% of correct classif.	% of correct r estimation				% of correct θ estimation			
		error tolerance ϵ_r				error tolerance ϵ_θ			
		± 0.125 cm	± 1 cm	± 5 cm	± 10 cm	$\pm 0.25^\circ$	$\pm 2^\circ$	$\pm 10^\circ$	$\pm 20^\circ$
plane	80	50	50	70	82	68	72	78	78
edge ($\theta_e = 60^\circ$)	88	60	60	68	84	68	76	80	84
cylinder ($r_c = 4.0$ cm)	76	60	60	76	86	70	76	76	76
cylinder ($r_c = 7.5$ cm)	74	62	62	72	80	66	70	70	72
cylinder ($r_c = 10.0$ cm)	74	62	64	76	84	70	72	72	72
average	78	59	59	72	83	68	73	75	76

Table 5.10: The percentages of correct classification, range (r) and azimuth (θ) estimation for the non-modular network trained with the back-propagation algorithm when the input signal I_2 is used for test set III.

test set	% of correct classif.	% of correct r estimation				% of correct θ estimation			
		error tolerance ϵ_r				error tolerance ϵ_θ			
		± 0.125 cm	± 1 cm	± 5 cm	± 10 cm	$\pm 0.25^\circ$	$\pm 2^\circ$	$\pm 10^\circ$	$\pm 20^\circ$
I	95(2.8)	73(4.2)	78(3.1)	88(1.7)	96(1.7)	90(2.8)	93(1.8)	97(1.1)	100(1.1)
II	92(3.2)	59(2.3)	61(2.3)	76(1.5)	85(2.0)	69(1.8)	73(1.4)	94(2.6)	94(1.2)
III	80(1.8)	59(2.9)	60(2.1)	73(1.8)	82(2.2)	69(1.7)	74(1.3)	76(2.3)	78(2.6)

Table 5.11: The mean and the standard deviation of the average percentages of correct classification, range (r) and azimuth (θ) estimation over ten non-modular networks trained with the back-propagation algorithm using different initial conditions for the connection weights. Input signal I_2 is used.

performances in target classification and localization with sonar. In most cases, the low-frequency component of the wavelet transform of the signal I_1 at resolution level $j = -2$ results in better classification and localization performance. Networks trained with the generating-shrinking algorithm demostrate better generalization capability compared to the networks trained with the back-propagation algorithm. When the results for non-modular and modular networks are compared, it is observed that the results for modular networks are in general slightly better than the results for non-modular ones. In the next chapter, statistical pattern recognition techniques are employed for target classification with sonar.

Chapter 6

STATISTICAL PATTERN RECOGNITION TECHNIQUES

In this chapter, application of statistical pattern recognition techniques to target classification with sonar are presented. The methods considered are two variations of the k-nearest neighbor method (k-NN), kernel estimation, parameterized density estimation, and linear discriminant analysis.

This chapter is organized as follows: statistical pattern recognition techniques are summarized in Section 6.1. In Section 6.2, these classification methods are applied to our problem.

6.1 Statistical Pattern Recognition Techniques

Classical classification problem can be defined as the classification of an object coming from class w_i if its corresponding vector representation $\mathbf{x}$ falls in the region Ω_i. A rule which partitions a space into regions Ω_i, $i = 1, \cdots, N$ where N is the number of possible classes is called a *decision rule*. Boundaries between regions are named

as *decision surfaces*. Let $p(w_i)$ be the probability of an object belonging to class w_i, $i = 1, \cdots, N$. Since these probabilities are independent of $\mathbf{x}$ and are known without making any observations, they are called *a priori probabilities*. Usually, they are taken equal to each other to minimize the probability of making an error. To classify an object with vector representation $\mathbf{x}$, *a posteriori probabilities* $p(w_i|\mathbf{x})$, $i = 1, \ldots, N$ (probabilities of belonging to each class) can be compared and the object is classified to class w_k if

$$p(w_k|\mathbf{x}) > p(w_i|\mathbf{x}) \text{ for all } i \neq k \Longrightarrow \mathbf{x} \in \Omega_k \tag{6.1}$$

The fundamental decision rule described above is known as *Bayes minimum error rule*. However, since these *a posteriori* probabilities are rarely known, they need to be estimated. A more convenient formulation of this rule can be obtained by using Bayes' theorem such that

$$p(w_i|\mathbf{x}) = \frac{p(\mathbf{x}|w_i)p(w_i)}{p(\mathbf{x})} \tag{6.2}$$

which results in

$$p(\mathbf{x}|w_k)p(w_k) > p(\mathbf{x}|w_i)p(w_i) \text{ for all } i \neq k \Longrightarrow \mathbf{x} \in \Omega_k \tag{6.3}$$

where $p(\mathbf{x}|w_i)$ are the class-conditional probability density functions. These class-conditional probability density functions are also unknown and need to be estimated. The set of vector representations used to estimate these class-conditional probability density functions is called the *design* or *training set*. The performance of any decision rule can be measured in a different set of vector representations which is called the *test set*. There are various statistical techniques to estimate these class-conditional density functions. These techniques are mainly divided into two as non-parametric and parametric. In non-parametric methods, no assumptions on the parametric form of the class-conditional densities are made. However, one of the main disadvantages of non-parametric estimation methods is that large data bases must be retained. This

is because it has been shown by Rosenblatt [134] that any non-parametric probability density function estimate based on a finite sample is biased. There are four major types of non-parametric probability density function estimators which are histogram, kernel estimator, k-nearest neighbor (k-NN) method, and series method. In parametric methods, parametric models for the class-conditional densities are assumed and then the parameters of these models are estimated. These parametric methods are mainly based on normal and non-normal models. In parametric methods based on the normal model, the parametric forms of the class-conditional densities are taken as the multi-variate normal density. In parametric methods based on a non-normal model, parametric form of the class-conditional densities can be taken as any suitable probability density function. The most commonly used estimation method in this case is the maximum likelihood estimator.

More generally, classification rules can be written as

$$q_k(\mathbf{x}) > q_i(\mathbf{x}) \text{ for all } i \neq k \Longrightarrow \mathbf{x} \in \Omega_k \tag{6.4}$$

where the function q_i is a *discriminant function.*

6.1.1 Kernel Estimator

Kernel estimator is a class of probability density estimator first proposed by Fix and Hodges in 1951 [135]. In the kernel estimator method, the class-conditional density function $p(\mathbf{x}|w_i)$ estimates are of the form

$$\hat{p}(\mathbf{x}|w_i) = \frac{1}{n_i h^d} \sum_{j=1}^{n_i} K\left(\frac{\mathbf{x} - \mathbf{x_j}}{h}\right) \tag{6.5}$$

where $\mathbf{x}$ is the d-dimensional vector at which the estimate is being made and $\mathbf{x_j}$'s, $j = 1, \cdots, n_i$ are the samples in the design set. In this equation, n_i is the total number of sample points in class w_i, h is called the *spread* or *smoothing parameter* or *bandwidth* of

a kernel estimator, and $K(\mathbf{z})$ is a kernel function which satisfies the conditions below:

$$K(\mathbf{z}) \geq 0 \tag{6.6}$$

$$\int K(\mathbf{z}) d\mathbf{z} = 1 \tag{6.7}$$

In this method, the selection of the smoothing parameter h is important. If h is selected too small, the estimator degenerates into a collection of n_i sharp peaks, each located at a sample point. On the other hand, if h is selected too large, the estimate is oversmoothed and an almost uniform probability density function results. Usually, h is chosen as a function of n_i such that

$$\lim_{n_i \to \infty} h(n_i) = 0 \tag{6.8}$$

Although any non-parametric estimate of $p(\mathbf{x}|w_i)$ based on a finite sample is biased, extra conditions on $K(\mathbf{z})$ can be imposed to ensure asymptotic unbiasedness. These are:

$$\int |K(\mathbf{z})| d\mathbf{z} < \infty \tag{6.9}$$

$$\sup_{-\infty < \mathbf{z} < \infty} |K(\mathbf{z})| < \infty \tag{6.10}$$

$$\lim_{\mathbf{z} \to \infty} |\mathbf{z} K(\mathbf{z})| = 0 \tag{6.11}$$

In addition, if h satisfies the condition $\lim_{n_i \to \infty} n_i h(n_i) = \infty$, then $\hat{p}(\mathbf{x}|w_i)$ is asymptotically consistent. Other properties of $\hat{p}(\mathbf{x}|w_i)$ under various conditions on $K(\mathbf{z})$ and h can be found in [136,137]. In the particular case of a Gaussian kernel, the estimate of $p(\mathbf{x}|w_i)$ can be expressed as:

$$\hat{p}(\mathbf{x}|w_i) = \frac{1}{n_i h^d} \sum_{j=1}^{n_i} \frac{1}{(2\pi)^{(d/2)} |\mathbf{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2h^2} (\mathbf{x} - \mathbf{x_j})^T \mathbf{\Sigma}^{-1} (\mathbf{x} - \mathbf{x_j}) \right] \tag{6.12}$$

Since the amount of computation involved in this estimate is excessive, frequently, uncorrelatedness of the elements of $\mathbf{x} - \mathbf{x_j}$ is assumed and $\mathbf{\Sigma}$ is replaced by a diagonal matrix. Then

$$\hat{p}(\mathbf{x}|w_i) = \frac{1}{n_i h^d} \sum_{j=1}^{n_i} \frac{1}{(2\pi)^{(d/2)} (\sigma_1 \cdots \sigma_d)^{1/2}} \prod_{m=1}^{d} \exp\left[-\frac{1}{2h^2} \frac{(x_m - x_{jm})^2}{\sigma_m^2} \right] \tag{6.13}$$

There are various approaches to select h if a constant h is to be used. Some of them can be summarized as:

- plot marginal estimates for several h values and choose one which is neither too smooth nor too degenerate
- find the average distance between sample points and their qth nearest neighbor and use this as h
- try different h values and select the one giving minimum misclassification rate by using the cross-validation approach. This approach can be summarized as follows: Each design set point is classified in turn based on the estimator designed using the other $n-1$ points in the design set, then the percentage of misclassification is calculated
- least-squares cross-validation in which h is chosen as minimizing integrated square error $\int [\hat{p}(\mathbf{x}|w_i) - p(\mathbf{x}|w_i)]^2 \, d\mathbf{x}$
- likelihood cross-validation in which h is chosen as maximizing the function $n^{-1} \sum_{k=1}^{n_i} \log \hat{p}_k(\mathbf{x_k}|w_i)$. Here $\hat{p}_k(\mathbf{x_k}|w_i)$ is the estimate constructed using all data points except $\mathbf{x_k}$ in the design set
- chose h which minimizes the mean squared error $\mathrm{MSE}_{\mathbf{x}}(\hat{p}(\mathbf{x}|w_i)) = E\{(\hat{p}(\mathbf{x}|w_i) - p(\mathbf{x}|w_i))\}^2$
- chose h which minimizes the mean integrated squared error $\mathrm{MISE}_{\mathbf{x}}(\hat{p}(\mathbf{x}|w_i)) = E\left\{\int [\hat{p}(\mathbf{x}|w_i) - p(\mathbf{x}|w_i)]^2 d\mathbf{x}\right\}$
- test graph method: plot second derivative of $\hat{p}(\mathbf{x}|w_i)$ for various h values and chose h resulting in rapid fluctuations which are quite marked but do not obscure the systematic variations completely.

For the further details of these approaches and for the other approaches, the reader can refer to the references [138–142].

6.1.2 k-Nearest Neighbor (k-NN) Method

For the class w_i with the class-conditional density function $p(\mathbf{x}|w_i)$, the probability that a point will fall in a local neighborhood L of $\mathbf{x}$ can be expressed as:

$$p_i(\mathbf{x}) = \int_{\mathbf{z}\in L} p(\mathbf{z}|w_i)d\mathbf{z} \tag{6.14}$$

If L is small and of volume V, then the above integral can be approximated as follows:

$$p_i(\mathbf{x}) \simeq p(\mathbf{x}|w_i)V \tag{6.15}$$

then

$$p(\mathbf{x}|w_i) \simeq p_i(\mathbf{x})/V \tag{6.16}$$

This approximation corresponds to the average value of $p(\mathbf{x}|w_i)$ in the region L around $\mathbf{x}$. From this approximation, a probability density function estimator can be developed as follows: $p_i(\mathbf{x})$ can be estimated simply by observing what proportion of the n_i sample points fall in L. If k_i is the number of sample points falling in L, then

$$\hat{p_i}(\mathbf{x}) = k_i/n_i \tag{6.17}$$

Therefore, the k-NN estimator for class w_i $\hat{p}(\mathbf{x}|w_i)$ can be defined as:

$$\hat{p}(\mathbf{x}|w_i) = \frac{k_i}{n_i V} \tag{6.18}$$

However, this estimator is not a valid probability density function since the integral of Equation (6.18) over the whole space is not unity but is infinity. Since the aim of the k-NN method is to classify new observation points, this method can be extended slightly by combining all the class sample points into one set of n points such that $\sum_{i=1}^{N} n_i = n$.

Let k be the number of points from the combined set that fall in V, k_i of it coming from class w_i. Then, the estimators for $p(w_i)$ and $p(\mathbf{x})$ can be defined as:

$$\hat{p}(w_i) = \frac{n_i}{n} \tag{6.19}$$

$$\hat{p}(\mathbf{x}) = \frac{k}{nV} \tag{6.20}$$

By using the Bayes' rule:

$$\hat{p}(w_i|\mathbf{x}) = \frac{\hat{p}(\mathbf{x}|w_i)\hat{p}(w_i)}{\hat{p}(\mathbf{x})} = \frac{k_i}{k} \tag{6.21}$$

Finally, this results in a classification rule such that $\mathbf{x}$ is classified as belonging to class w_m if $k_m = \max_i(k_i)$. Therefore, an object with vector representation $\mathbf{x}$ in the test set is classified as belonging to a class w_m simply by voting among the classes of the vector representations of k nearest neighbors of $\mathbf{x}$ in the training set where w_m is the class which receives the maximum vote. One of the main disadvantages of this method is that a predefined rule for the selection of the k value does not exist.

Another interpretation of the k-NN estimator which relates it to the kernel estimator can be found in [139]. Let $r_k(\mathbf{x})$ be the Euclidean distance from $\mathbf{x}$ to the kth nearest neighbor of $\mathbf{x}$ and $V_k(\mathbf{x})$ be the volume of the d-dimensional sphere of radius $r_k(\mathbf{x})$. Therefore,

$$V_k(\mathbf{x}) = c_d[r_k(\mathbf{x})]^d \tag{6.22}$$

and

$$\hat{p}(\mathbf{x}|w_i) = \frac{k}{n_i V_k(\mathbf{x})} = \frac{k}{n_i c_d[r_k(\mathbf{x})]^d}. \tag{6.23}$$

Here c_d is volume of the unit sphere in d-dimensional space. Now, consider the kernel

$$K(\mathbf{z}) = \begin{cases} c_d^{-1} & \text{if } |\mathbf{x}| \leq 1 \\ 0 & \text{otherwise.} \end{cases} \tag{6.24}$$

Then, the kernel estimate of $\hat{p}(\mathbf{x}|w_i)$ with the smoothing parameter $h = r_k(\mathbf{x})$ is equal to the estimate of $\hat{p}(\mathbf{x}|w_i)$ which is given in Equation (6.23). Finally, this relation can be extended to a general kernel such that

$$\hat{p}(\mathbf{x}|w_i) = \frac{1}{n_i[r_k(\mathbf{x})]^d} \sum_{j=1}^{n_i} K\left(\frac{\mathbf{x} - \mathbf{x_j}}{r_k(\mathbf{x})}\right). \tag{6.25}$$

This estimator is referred as *generalized k-nearest neighbor* estimator. The major difference between the kernel estimator and the generalized k-NN estimator is that in the generalized k-NN estimator, different smoothing parameter values $r_k(\mathbf{x})$ are used for each vector representation $\mathbf{x}$ instead of a constant smoothing parameter h for all vector representations as in the kernel estimator.

6.1.3 Parameterized Density Estimation with Normal Models

In this method, each class-conditional density function is assumed to be a multivariate normal such that

$$p(\mathbf{x}|\mathbf{x_1}, \cdots, \mathbf{x_{n_i}}) = \frac{1}{(2\pi)^{(d/2)}|\mathbf{\Sigma_i}|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \mu_\mathbf{i})^T \mathbf{\Sigma_i}^{-1}(\mathbf{x} - \mu_\mathbf{i})\right], \; i = 1, \cdots, N \tag{6.26}$$

where $\mu_\mathbf{i}$'s denote the class mean, and $\mathbf{\Sigma_i}$'s denote the class-covariance matrices. The parameters $\theta_\mathbf{i} = (\mu_\mathbf{i}, \mathbf{\Sigma_i})$ must be estimated by using estimation techniques based on the design set. The most commonly used estimation technique is the maximum likelihood estimation. In maximum likelihood estimation, a *likelihood function* which is maximized with respect to the parameters to be estimated is defined as:

$$\mathcal{L}(\mathbf{x_1}, \cdots, \mathbf{x_{n_i}}|\theta_\mathbf{i}) = \prod_{j=1}^{n_i} p(\mathbf{x_j}|\theta_\mathbf{i}) \tag{6.27}$$

Then,

$$\mathcal{L}(\mathbf{x_1}, \cdots, \mathbf{x_{n_i}}|\theta_\mathbf{i}) = \frac{1}{(2\pi)^{(dn_i/2)}|\mathbf{\Sigma_i}|^{n_i/2}} \exp\left\{-\frac{1}{2}\sum_{j=1}^{n_i}\left[(\mathbf{x_j} - \mu_\mathbf{i})^T \mathbf{\Sigma_i}^{-1}(\mathbf{x_j} - \mu_\mathbf{i})\right]\right\} \tag{6.28}$$

Clearly, any monotonic transformation of $\mathcal{L}$ has its maximum at the same $\theta_\mathbf{i}$ value as $\mathcal{L}$ does. Natural logarithm of $\mathcal{L}$ is frequently used. To find the $\theta_\mathbf{i}$ value which maximizes $\mathcal{L}$, one can differentiate $\ln \mathcal{L}$ with respect to $\theta_\mathbf{i}$, equate the resulting expressions to zero to give the *normal* equations and finally solve these equations for $\hat{\theta}_\mathbf{i}$. Through slightly more complicated algebra [143], the maximum likelihood estimates of $\mu_\mathbf{i}$'s and $\mathbf{\Sigma_i}$'s can be derived as:

$$\hat{\mu_\mathbf{i}} = \frac{1}{n_i} \sum_{j=1}^{n_i} \mathbf{x_j} \tag{6.29}$$

$$\hat{\mathbf{\Sigma}}_\mathbf{i} = \frac{1}{n_i} \sum_{j=1}^{n_i} (\mathbf{x_j} - \mu_\mathbf{i})(\mathbf{x_j} - \mu_\mathbf{i})^T \tag{6.30}$$

Normal models used with parameterized density estimation are divided into two groups which are namely *heteroscedastic* normal model and *homoscedastic* normal model. In the homoscedastic normal model, all class-covariance matrices are equal ($\hat{\mathbf{\Sigma}}_\mathbf{i} = \mathbf{\Sigma}$ for all $i = 1, \cdots, N$). Usually $\mathbf{\Sigma}$ is taken as the weighted average of each class-covariance matrix estimate $\hat{\mathbf{\Sigma}}_\mathbf{i}$ [144] such that

$$\mathbf{\Sigma} = \sum_{i=1}^{N} \frac{n_i}{n} \hat{\mathbf{\Sigma}}_\mathbf{i} \tag{6.31}$$

In the heteroscedastic normal model, different class-covariance matrices are used for each class.

6.1.4 Linear Discriminant Analysis

In linear discriminant analysis, discriminant functions $q(\mathbf{x})$'s are linear functions of x_i such that

$$q(\mathbf{x}) = a_\circ + \sum_{i=1}^{d} a_i x_i = \mathbf{a}^T \mathbf{z} \tag{6.32}$$

where $\mathbf{z} = (1, \mathbf{x}^T)^T$ is the augmented observation vector. Our aim is to find the *weight* vector $\mathbf{a}$, based on the design set which consists of two class samples such that

i) $\mathbf{a}^T\mathbf{z_i} > 0$, whenever $\mathbf{x_i}$ is a sample from class w_1, and

ii) $\mathbf{a}^T\mathbf{z_i} < 0$, whenever $\mathbf{x_i}$ is a sample from class w_2

Without loss of generality, one can define $\mathbf{y_i}$ such that

i) $\mathbf{y_i} = \mathbf{z_i}$, whenever $\mathbf{x_i}$ is a sample from class w_1, and

ii) $\mathbf{y_i} = -\mathbf{z_i}$, whenever $\mathbf{x_i}$ is a sample from class w_2

Now, the weight vector $\mathbf{a}$ must satisfy the condition $\mathbf{a}^T\mathbf{y_i} > 0$ for all $\mathbf{y_i}$ corresponding to $\mathbf{x_i}$ in the design set. The decision surface estimated from this discriminant function is $\mathbf{a}^T\mathbf{y_i} = 0$ which is a hyperplane. In fact, unless these two classes are linearly separable, a weight vector $\mathbf{a}$ which satisfies the above condition cannot be found. Therefore, the aim of this classification process is to satisfy $\mathbf{a}^T\mathbf{y_i} > 0$ for as many of the sample points as possible by minimizing the *misclassification rate*. However, the misclassification rate is a difficult criterion to minimize. There are various criteria to find the linear surface which best discriminates two classes. Two of the most widely used ones are perceptron criterion and Fisher's criterion. The perceptron criterion minimizes

$$C_1(\mathbf{a}) = \sum_M (-\mathbf{a}^T\mathbf{y_i}), \text{ where } M = \mathbf{y_i} \text{ such that } \mathbf{a}^T\mathbf{y_i} < 0 \tag{6.33}$$

Fisher's criterion maximizes of the ratio of the distance between sample means to the standard deviation within samples. That is

$$C_2(\mathbf{a}) = \frac{\mathbf{a}^T\bar{\mathbf{x}}_1 - \mathbf{a}^T\bar{\mathbf{x}}_2}{\sqrt{\mathbf{a}^T\mathbf{S}\mathbf{a}}} \tag{6.34}$$

A weight vector $\mathbf{a}$ that satisfies $\mathbf{a}^T\mathbf{y_i} = b_i$ can be found instead of finding the weight vector $\mathbf{a}$ such that as many of the sample points as possible satisfy $\mathbf{a}^T\mathbf{y_i} > 0$. Here, b_i's are positive constants for the sample points. Then n equations in matrix form $\mathbf{a}^T\mathbf{Y} = \mathbf{B}$ must be solved where $\mathbf{Y} = (\mathbf{y_1}, \mathbf{y_2}, \cdots, \mathbf{y_n})$ and $\mathbf{B} = (b_1 \cdots b_n)^T$. For a given $\mathbf{B}$, the least-squares approach which minimizes

$$C_3(\mathbf{a}) = (\mathbf{a}^T\mathbf{Y} - \mathbf{B})^T(\mathbf{a}^T\mathbf{Y} - \mathbf{B}) \tag{6.35}$$

results in the solution

$$\mathbf{a} = (\mathbf{Y}\mathbf{Y}^T)^{-1}\mathbf{Y}\mathbf{B} \tag{6.36}$$

An interesting choice for $\mathbf{B}$ is

$$\mathbf{B} = \begin{bmatrix} \frac{n}{n_1}\mathbf{u_1} \\ \frac{n}{n_2}\mathbf{u_2} \end{bmatrix} \tag{6.37}$$

where $\mathbf{u_i}$ is a vector of n_i ones, n_i is the number of design set vectors from class w_i, and $n_1 + n_2 = n$. Least-squares approach with this choice of $\mathbf{B}$ results in exactly the same solution obtained with Fisher's criterion [138].

Generalization of linear discriminant analysis to N classes can be done in several ways summarized below:

- use $N-1$ two-class decision rules, the first separating Ω_1 from $\Omega_2, \cdots, \Omega_N$, the second separating Ω_2 from $\Omega_3, \cdots, \Omega_N$, the third separating Ω_3 from $\Omega_4, \cdots, \Omega_N$, etc.
- use $N-1$ two-class decision rules, each one separating $\Omega_i,\ i = 1, \cdots, N-1$ from *all* $\Omega_j\ j = 1, \cdots, N$ where $j \neq i$
- use $N(N-1)/2$ two-class decision rules, one for each pair of classes

6.2 Experimental Studies

Again, the same ultrasonic sensor node used in all our previous studies is employed with step size 1.8° (Figure 2.10). The same training set is used for designing the classification rule. The training set is generated by scanning the following targets with the sensor node, while the targets are located at the 25 sensing locations of Figure 4.1: cylinders with radii 2.5, 5.0 and 7.5 cm, a planar target, a corner, an edge of $\theta_e = 90°$, and an acute corner of $\theta_c = 60°$.

The same vector representations used with fuzzy c-means clustering algorithm described in Section 4.3 are used to construct three different design sets. Note that the entries of these vector representations are the samples of the corresponding signals I_1, I_2, and I_3 which are used as the input signals to the neural network classifiers in Section 5.5. Three different test sets I, II, and III are used to test each classification method with each design set. These test sets are the same test sets used throughout this thesis.

First, the k-NN and generalized k-NN methods are used to classify the vector representations of target types in each test set with the corresponding design set for k values varying between 1 and 10. The resulting percentages of correct classification for each target type and their averages are tabulated for the three different vector representations and three test sets in Tables 6.1–6.9. In all of the tables, the numbers before the parentheses are the testing results for k-NN, whereas the numbers given in parentheses are those for generalized k-NN. For the k-NN method, the highest average percentages of correct classification in testing results of test set I are 97%, 98% and 91% for the vector representations $\mathbf{X_I}$, $\mathbf{X_{II}}$, and $\mathbf{X_{III}}$, respectively. For test set II, the highest average percentages of correct classification are 83%, 74%, and 67%, and those for test set III are 70%, 67%, and 63% for the vector representations $\mathbf{X_I}$, $\mathbf{X_{II}}$, and $\mathbf{X_{III}}$, respectively. The average percentages of correct classification decreases with increasing k values for all test sets and vector representations. For vector representations $\mathbf{X_I}$ and $\mathbf{X_{II}}$, the average percentages of correct classification are comparable up to $k = 6$. After this k value, the results for $\mathbf{X_I}$ is better than those for $\mathbf{X_{II}}$ for test set I. Moreover, the average percentages of correct classification for $\mathbf{X_I}$ is always higher than those for $\mathbf{X_{II}}$ for test sets II and III. The results for the third vector representation are not satisfactory compared to the results obtained with the first two vector representations for all test sets.

target type \ k	1	2	3	4	5	6	7	8	9	10
plane	100(100)	100(100)	100(100)	100(100)	100(100)	100(100)	100(100)	100(100)	100(100)	100(100)
corner	100(100)	100(100)	100(100)	100(100)	100(100)	100(100)	100(100)	100(100)	100(100)	100(100)
edge ($\theta_e = 90°$)	100(100)	100(100)	100(100)	100(100)	100(100)	100(100)	100(100)	100(100)	100(100)	100(100)
acute corner ($\theta_c = 60°$)	99(100)	99(100)	99(100)	99(100)	99(96)	99(96)	99(96)	99(96)	99(99)	95(97)
cylinder ($r_c = 2.5$ cm)	96(99)	96(99)	96(99)	97(99)	93(92)	93(95)	93(95)	88(96)	88(71)	88(73)
cylinder ($r_c = 5.0$ cm)	92(96)	95(96)	91(96)	94(96)	91(61)	89(59)	84(61)	86(61)	81(50)	83(45)
cylinder ($r_c = 7.5$ cm)	93(100)	89(100)	92(100)	85(100)	89(57)	85(56)	83(56)	61(59)	67(62)	63(65)
average	97(99)	97(99)	97(99)	96(99)	96(87)	95(87)	94(87)	91(87)	91(83)	90(83)

Table 6.1: The percentages of correct classification when k-NN and generalized k-NN methods are employed in test set I with k values between 1 and 10 for vector representation $\mathbf{X_I}$.

target type \ k	1	2	3	4	5	6	7	8	9	10
plane	100(100)	100(100)	100(100)	100(100)	100(83)	100(83)	95(83)	95(83)	85(80)	85(80)
corner	100(100)	100(100)	100(100)	100(96)	98(93)	98(89)	98(88)	98(88)	98(84)	98(84)
edge ($\theta_e = 90°$)	100(100)	98(100)	97(100)	97(100)	97(100)	97(100)	97(100)	95(100)	92(100)	92(100)
acute corner ($\theta_c = 60°$)	98(100)	97(100)	98(100)	96(100)	97(96)	96(96)	93(96)	87(96)	89(100)	90(100)
cylinder ($r_c = 2.5$ cm)	96(98)	96(99)	96(99)	96(98)	92(77)	93(76)	92(74)	93(76)	88(61)	87(60)
cylinder ($r_c = 5.0$ cm)	94(95)	96(95)	92(95)	94(95)	94(44)	94(38)	90(33)	86(26)	79(30)	80(31)
cylinder ($r_c = 7.5$ cm)	95(100)	92(100)	95(100)	87(100)	86(67)	81(65)	82(63)	60(60)	68(56)	64(58)
average	98(99)	97(99)	97(99)	96(98)	95(80)	94(78)	92(77)	88(76)	86(73)	85(73)

Table 6.2: The percentages of correct classification when k-NN and generalized k-NN methods are employed in test set I with k values between 1 and 10 for vector representation $\mathbf{X_{II}}$.

target type \ k	1	2	3	4	5	6	7	8	9	10
plane	92(100)	93(100)	88(100)	86(100)	78(84)	79(84)	66(82)	65(81)	61(80)	55(80)
corner	97(100)	99(100)	99(100)	99(92)	99(71)	99(71)	99(71)	99(71)	96(67)	96(67)
edge ($\theta_e = 90°$)	100(100)	96(100)	94(100)	92(99)	92(100)	87(100)	79(100)	80(100)	79(100)	76(100)
acute corner ($\theta_c = 60°$)	98(100)	96(100)	96(100)	95(100)	97(92)	95(92)	95(92)	91(91)	89(90)	88(88)
cylinder ($r_c = 2.5$ cm)	89(99)	88(99)	81(96)	84(95)	81(78)	83(75)	79(75)	70(72)	77(52)	75(53)
cylinder ($r_c = 5.0$ cm)	80(93)	76(94)	71(91)	72(91)	64(58)	61(55)	59(54)	57(51)	48(22)	54(22)
cylinder ($r_c = 7.5$ cm)	82(99)	68(100)	74(100)	67(100)	62(57)	55(59)	52(57)	51(56)	53(34)	48(35)
average	91(99)	88(99)	86(98)	85(97)	82(77)	80(77)	76(76)	73(75)	72(64)	70(64)

Table 6.3: The percentages of correct classification when k-NN and generalized k-NN methods are employed in test set I with k values between 1 and 10 for vector representation $\mathbf{X_{III}}$.

Higher percentages of correct classification are obtained with the generalized k-NN method than with k-NN. For the generalized k-NN method, the highest average percentage of correct classification achieved with test set I is 99% for all vector representations. For test set II, the highest average percentages of correct classification are 95%, 90%, and 83%, and for test set III 71%, 69%, and 67% for the vector

k / target type	1	2	3	4	5	6	7	8	9	10
plane	100(100)	100(100)	100(100)	100(100)	100(100)	100(100)	100(100)	100(100)	100(100)	100(100)
corner	97(100)	100(100)	100(100)	100(100)	100(100)	100(100)	100(100)	100(100)	100(100)	100(100)
edge ($\theta_e = 90°$)	100(100)	100(100)	100(100)	100(100)	100(100)	100(100)	100(100)	100(100)	100(100)	100(100)
acute corner ($\theta_c = 60°$)	83(97)	83(97)	83(97)	83(97)	83(95)	83(95)	83(95)	80(95)	83(93)	81(92)
cylinder ($r_c = 2.5$ cm)	71(100)	71(97)	65(97)	65(93)	63(93)	67(95)	67(95)	67(92)	74(79)	74(80)
cylinder ($r_c = 5.0$ cm)	79(88)	81(85)	79(88)	80(88)	79(58)	75(57)	72(58)	73(61)	74(55)	75(53)
cylinder ($r_c = 7.5$ cm)	50(77)	48(77)	53(77)	53(80)	51(52)	56(48)	58(48)	41(53)	47(58)	38(60)
average	83(95)	83(94)	83(94)	83(94)	82(86)	83(85)	83(85)	80(86)	83(84)	81(84)

Table 6.4: The percentages of correct classification when k-NN and generalized k-NN methods are employed in test set II with k values between 1 and 10 for vector representation $\mathbf{X_I}$.

k / target type	1	2	3	4	5	6	7	8	9	10
plane	60(87)	64(87)	60(87)	60(87)	60(65)	64(65)	61(65)	58(65)	53(64)	50(64)
corner	94(100)	97(100)	97(97)	97(95)	96(87)	96(85)	93(84)	99(84)	93(86)	93(86)
edge ($\theta_e = 90°$)	87(100)	69(100)	72(100)	72(100)	72(100)	72(100)	72(100)	58(100)	56(100)	56(100)
acute corner ($\theta_c = 60°$)	73(100)	72(100)	76(100)	75(100)	76(95)	72(95)	70(95)	67(95)	72(97)	72(97)
cylinder ($r_c = 2.5$ cm)	78(89)	78(90)	78(90)	78(89)	76(82)	80(82)	80(81)	87(78)	81(71)	77(70)
cylinder ($r_c = 5.0$ cm)	74(81)	72(75)	70(78)	74(75)	71(36)	74(36)	72(34)	70(30)	70(29)	70(29)
cylinder ($r_c = 7.5$ cm)	51(70)	50(70)	58(70)	51(74)	50(61)	51(53)	51(52)	44(54)	48(55)	42(56)
average	74(90)	72(89)	73(89)	72(89)	72(75)	73(74)	71(73)	69(73)	68(72)	66(72)

Table 6.5: The percentages of correct classification when k-NN and generalized k-NN methods are employed in test set II with k values between 1 and 10 for vector representation $\mathbf{X_{II}}$.

k / target type	1	2	3	4	5	6	7	8	9	10
plane	49(84)	50(84)	44(84)	43(84)	39(62)	40(62)	33(61)	33(61)	31(60)	28(60)
corner	85(87)	96(87)	93(87)	96(83)	96(69)	96(69)	93(69)	100(69)	91(64)	95(64)
edge ($\theta_e = 90°$)	93(97)	68(97)	67(97)	66(97)	63(97)	57(97)	53(97)	50(97)	50(97)	48(97)
acute corner ($\theta_c = 60°$)	69(94)	65(97)	65(97)	64(97)	65(90)	61(90)	61(90)	59(89)	58(89)	57(88)
cylinder ($r_c = 2.5$ cm)	65(80)	67(77)	57(75)	59(78)	57(73)	58(75)	60(65)	62(66)	62(50)	64(47)
cylinder ($r_c = 5.0$ cm)	57(74)	48(77)	42(76)	43(69)	39(46)	37(45)	36(44)	35(49)	34(25)	37(21)
cylinder ($r_c = 7.5$ cm)	48(57)	37(57)	50(64)	50(67)	51(42)	41(47)	33(46)	32(45)	33(34)	31(35)
average	67(82)	62(82)	60(83)	60(82)	59(69)	56(69)	53(67)	53(68)	51(60)	51(59)

Table 6.6: The percentages of correct classification when k-NN and generalized k-NN methods are employed in test set II with k values between 1 and 10 for vector representation $\mathbf{X_{III}}$.

representations $\mathbf{X_I}$, $\mathbf{X_{II}}$, and $\mathbf{X_{III}}$, respectively. For test set I, highest average percentages of correct classification in the range of 97−99% are obtained for $1 \leq k \leq 4$ for all three vector representations. When $k > 4$, the percentages decrease with increasing k values for all three vector representations and the classification results for $\mathbf{X_I}$ are always better than those for $\mathbf{X_{II}}$ and $\mathbf{X_{III}}$. For test sets II and III, the highest average

target type \ k	1	2	3	4	5	6	7	8	9	10
plane	73(73)	73(73)	73(72)	73(71)	72(68)	73(67)	71(67)	75(67)	68(66)	68(66)
edge ($\theta_e = 60°$)	96(96)	96(96)	96(96)	96(96)	96(96)	96(96)	96(96)	96(96)	96(96)	96(96)
cylinder ($r_c = 4.0$ cm)	57(58)	56(60)	58(63)	59(64)	60(59)	60(62)	61(62)	56(63)	58(48)	59(49)
cylinder ($r_c = 7.5$ cm)	54(56)	54(56)	54(58)	55(59)	53(50)	55(52)	57(54)	49(54)	50(48)	49(52)
cylinder ($r_c = 10.0$ cm)	64(66)	64(67)	65(66)	65(67)	63(64)	63(66)	64(64)	56(65)	62(53)	61(55)
average	69(70)	69(70)	69(71)	70(71)	69(67)	69(69)	70(69)	66(69)	67(62)	67(64)

Table 6.7: The percentages of correct classification when k-NN and generalized k-NN methods are employed in test set III with k values between 1 and 10 for vector representation $\mathbf{X_I}$.

target type \ k	1	2	3	4	5	6	7	8	9	10
plane	68(68)	68(68)	68(68)	68(68)	68(60)	68(60)	66(60)	66(60)	61(57)	61(56)
edge ($\theta_e = 60°$)	94(94)	95(96)	95(96)	95(96)	95(95)	95(94)	95(94)	94(96)	92(96)	92(96)
cylinder ($r_c = 4.0$ cm)	57(58)	56(58)	57(61)	56(60)	56(52)	58(50)	57(50)	57(51)	57(39)	58(33)
cylinder ($r_c = 7.5$ cm)	53(54)	53(55)	53(55)	53(53)	51(44)	51(43)	50(42)	49(43)	49(38)	48(37)
cylinder ($r_c = 10.0$ cm)	60(61)	59(62)	60(63)	60(62)	59(52)	60(50)	59(48)	55(51)	54(46)	53(43)
average	66(67)	66(68)	67(69)	66(68)	66(61)	66(59)	65(59)	64(60)	63(55)	62(53)

Table 6.8: The percentages of correct classification when k-NN and generalized k-NN methods are employed in test set III with k values between 1 and 10 for vector representation $\mathbf{X_{II}}$.

target type \ k	1	2	3	4	5	6	7	8	9	10
plane	68(72)	69(72)	66(72)	65(72)	61(62)	62(61)	55(60)	56(60)	54(58)	51(58)
edge ($\theta_e = 60°$)	93(93)	91(93)	90(92)	89(93)	89(93)	87(93)	83(93)	83(93)	83(93)	81(93)
cylinder ($r_c = 4.0$ cm)	52(57)	51(58)	48(58)	49(57)	49(50)	51(49)	49(51)	43(47)	49(34)	48(34)
cylinder ($r_c = 7.5$ cm)	46(51)	44(52)	43(49)	43(48)	42(41)	42(42)	40(42)	35(41)	41(35)	40(37)
cylinder ($r_c = 10.0$ cm)	54(59)	51(59)	50(60)	51(58)	53(50)	52(49)	50(49)	40(47)	46(39)	47(40)
average	63(66)	61(67)	59(66)	59(66)	59(59)	59(59)	55(59)	51(58)	55(52)	53(52)

Table 6.9: The percentages of correct classification when k-NN and generalized k-NN methods are employed in test set III with k values between 1 and 10 for vector representation $\mathbf{X_{III}}$.

percentages of correct classification are obtained for vector representation $\mathbf{X_I}$, followed by vector representations $\mathbf{X_{II}}$ and $\mathbf{X_{III}}$ in the given order. For both k-NN and generalized k-NN estimators, the correct classification rates for test set II and III are lower than those for test set I for most cases. Moreover, the correct classification rates for test set II are better than those for test set III for most cases.

Next, kernel estimators are designed based on each design set. In these estimators, Gaussian kernels are used and the smoothing parameter h is found by finding the average

distances between sample points and their qth nearest neighbor for $1 \leq q \leq 10$, and selecting the average distance giving minimum misclassification rate by using the cross-validation approach. For all the vector representations, the average distances for $q = 4$ which gives the minimum misclassification rate in cross-validation approach is selected. The resulting percentages of correct classification for each target type and their averages are tabulated for the three vector representations in Table 6.10 for test set I and II. In this table, the numbers before the parentheses are for test set I, whereas the numbers given in parentheses are the results of test set II. For test set III, the resulting percentages are tabulated in Table 6.11. The results with kernel estimators are always better than the results obtained with k-NN method. However, for the vector representation $\mathbf{X_{III}}$ the results with generalized k-NN when $1 \leq k \leq 4$ are always better than the results obtained with the kernel estimator while the results with generalized k-NN when $1 \leq k \leq 4$ are comparable with the results of kernel estimator for the vector representations $\mathbf{X_I}$ and $\mathbf{X_{II}}$ for test set I. When $1 \leq k \leq 4$ the results with generalized k-NN are always better than the results obtained with the kernel estimator for all three vector representations for test set II, while they are comparable for test set III. When $k > 4$, the situation is reversed for all test sets. For test set I, the average percentages of correct classification obtained with kernel estimator are 99%, 99%, and 94% for the vector representations $\mathbf{X_I}$, $\mathbf{X_{II}}$, and $\mathbf{X_{III}}$, respectively. For test set II, these percentages are 93%, 89%, and 78% and those for test set III are 71%, 68%, and 65% for the vector representations $\mathbf{X_I}$, $\mathbf{X_{II}}$, and $\mathbf{X_{III}}$, respectively.

Thirdly, parameterized density estimation with the normal model is used to estimate class-conditional probability density function for each class (i.e., target type) in each design set. Heteroscedastic and homoscedastic normal models are employed. The resulting percentages of correct classification for each target type and their averages for test sets I and II are tabulated for the three vector representations with heteroscedastic and homoscedastic normal models in Tables 6.12 and 6.13, respectively. In these tables, the numbers before the parentheses are for test set I, whereas the numbers given in

vector representation / target type	$\mathbf{X_I}$	$\mathbf{X_{II}}$	$\mathbf{X_{III}}$
plane	100(100)	100(87)	100(83)
corner	100(100)	100(100)	100(87)
edge ($\theta_e = 90°$)	100(100)	100(100)	100(97)
acute corner ($\theta_c = 60°$)	100(97)	100(100)	100(93)
cylinder ($r_c = 2.5$ cm)	95(94)	99(90)	77(59)
cylinder ($r_c = 5.0$ cm)	95(88)	95(71)	84(65)
cylinder ($r_c = 7.5$ cm)	99(93)	100(77)	94(78)
average	99(93)	99(89)	94(78)

Table 6.10: The percentages of correct classification when kernel estimator is employed in test sets I and II for the three vector representations $\mathbf{X_I}$, $\mathbf{X_{II}}$, and $\mathbf{X_{III}}$.

vector representation / target type	$\mathbf{X_I}$	$\mathbf{X_{II}}$	$\mathbf{X_{III}}$
plane	73	68	72
edge ($\theta_e = 90°$)	96	96	94
cylinder ($r_c = 4.0$ cm)	65	61	52
cylinder ($r_c = 7.5$ cm)	55	54	49
cylinder ($r_c = 10.0$ cm)	65	63	58
average	71	68	65

Table 6.11: The percentages of correct classification when kernel estimator is employed in test set III for the three vector representations $\mathbf{X_I}$, $\mathbf{X_{II}}$, and $\mathbf{X_{III}}$.

parentheses are the results of test set II. The testing results of test set III for both heteroscedastic and homoscedastic normal models are given in Table 6.14. In this table, the numbers before the parentheses are for heteroscedastic normal model, whereas the numbers given in parentheses are the results of the homoscedastic normal model. Referring to these tables, on the average, the percentages of correct classification obtained with the heteroscedastic normal model are slightly higher than those obtained with the homoscedastic normal model. Nevertheless, the percentages obtained with these models are always lower than those obtained with both the k-NN methods and the

kernel estimator for the three test sets. This result can be expected since both of these methods are non-parametric in which no assumptions on the underlying density functions for each class are made. On the other hand, in parameterized density estimation with the normal model, the class-conditional densities are assumed to have Gaussian distribution which imposes an unnecessary restriction.

vector representation / target type	$\mathbf{X_I}$	$\mathbf{X_{II}}$	$\mathbf{X_{III}}$
plane	100(100)	96(78)	96(78)
corner	87(81)	99(97)	80(80)
edge ($\theta_e = 90°$)	98(99)	83(92)	65(70)
acute corner ($\theta_c = 60°$)	100(94)	96(88)	73(67)
cylinder ($r_c = 2.5$ cm)	64(56)	42(31)	52(43)
cylinder ($r_c = 5.0$ cm)	42(38)	64(56)	74(81)
cylinder ($r_c = 7.5$ cm)	76(82)	70(75)	36(28)
average	81(79)	79(74)	68(64)

Table 6.12: The percentages of correct classification when parameterized density estimation with heteroscedastic normal model employed in test set I and II for the three vector representations $\mathbf{X_I}$, $\mathbf{X_{II}}$, and $\mathbf{X_{III}}$.

vector representation / target type	$\mathbf{X_I}$	$\mathbf{X_{II}}$	$\mathbf{X_{III}}$
plane	96(95)	88(71)	84(69)
corner	92(86)	58(52)	46(33)
edge ($\theta_e = 90°$)	100(100)	97(99)	78(79)
acute corner ($\theta_c = 60°$)	69(68)	72(63)	48(47)
cylinder ($r_c = 2.5$ cm)	68(77)	85(69)	74(70)
cylinder ($r_c = 5.0$ cm)	44(35)	45(33)	44(32)
cylinder ($r_c = 7.5$ cm)	65(56)	55(58)	57(45)
average	76(74)	71(63)	62(54)

Table 6.13: The percentages of correct classification when parameterized density estimation with homoscedastic normal model employed in test set I and II for the three vector representations $\mathbf{X_I}$, $\mathbf{X_{II}}$, and $\mathbf{X_{III}}$.

vector representation / target type	$\mathbf{X_I}$	$\mathbf{X_{II}}$	$\mathbf{X_{III}}$
plane	85(57)	74(64)	71(60)
edge ($\theta_e = 60°$)	94(100)	80(93)	65(76)
cylinder ($r_c = 4.0$ cm)	44(38)	45(47)	62(48)
cylinder ($r_c = 7.5$ cm)	51(44)	48(48)	52(45)
cylinder ($r_c = 10.0$ cm)	57(48)	49(53)	59(53)
average	66(57)	59(61)	62(56)

Table 6.14: The percentages of correct classification when parameterized density estimation with heteroscedastic and homoscedastic normal models employed in test set III for the three vector representations $\mathbf{X_I}$, $\mathbf{X_{II}}$, and $\mathbf{X_{III}}$.

Finally, linear discriminant analysis is employed to classify the seven target types. In this analysis, the least-squares approach is used with $\mathbf{B}$ matrices which are chosen according to Equation (6.37), and $N-1$ two-class decision rules, each one separating $\Omega_i,\ i = 1, \cdots, N-1$ from all $\Omega_j\ j = 1, \cdots, N,\ j \neq i$ are designed. In this study, $N = 7$ and $n_i = 100,\ i = 1, \cdots, 7$ for each design set. The resulting percentages of correct classification for each target type and their averages for test set I and II are tabulated for the three vector representations in Table 6.15. In this table, the numbers before the parentheses are for test set I, whereas the numbers given in parentheses are the results of test set II. The results of test set III are given in Table 6.16. Referring to these tables, it can be observed that these results are even worse than the results obtained by employing parameterized density estimation with the homoscedastic normal model. These results indicate that the three vector representations of the data collected from the target types are not suitable for linear separation.

The results obtained with vector representation $\mathbf{X_I}$ are better than the results obtained with the other vector representations, whereas vector representation $\mathbf{X_{III}}$ results in the lowest percentages of correct classification among all vector representations for all methods considered in this chapter. The best classification performance for all methods considered in this chapter is obtained with test set I, followed by test sets II and III

vector representation / target type	$\mathbf{X_I}$	$\mathbf{X_{II}}$	$\mathbf{X_{III}}$
plane	83(67)	52(30)	49(28)
corner	58(47)	56(33)	51(34)
edge ($\theta_e = 90°$)	74(69)	59(36)	81(72)
acute corner ($\theta_c = 60°$)	54(39)	53(29)	47(30)
cylinder ($r_c = 2.5$ cm)	70(57)	59(42)	68(59)
cylinder ($r_c = 5.0$ cm)	83(60)	70(59)	53(45)
cylinder ($r_c = 7.5$ cm)	76(52)	58(43)	47(27)
average	71(56)	58(39)	57(42)

Table 6.15: The percentages of correct classification when linear discriminant analysis is employed in test sets I and II for the three vector representations $\mathbf{X_I}$, $\mathbf{X_{II}}$, and $\mathbf{X_{III}}$.

vector representation / target type	$\mathbf{X_I}$	$\mathbf{X_{II}}$	$\mathbf{X_{III}}$
plane	49	35	32
edge ($\theta_e = 60°$)	63	45	55
cylinder ($r_c = 4.0$ cm)	48	43	44
cylinder ($r_c = 7.5$ cm)	43	42	34
cylinder ($r_c = 10.0$ cm)	48	40	38
average	50	41	41

Table 6.16: The percentages of correct classification when linear discriminant analysis is employed in test set III for the three vector representations $\mathbf{X_I}$, $\mathbf{X_{II}}$, and $\mathbf{X_{III}}$.

in the given order. Moreover, non-parametric methods (i.e., k-NN methods and kernel estimator) always outperform parameterized density estimation and linear discriminant analysis.

In this chapter, application of various statistical pattern recognition techniques to target classification is presented. In the next chapter, all approaches used for target classification and localization with sonar throughout this thesis are compared experimentally in a common test pool.

Chapter 7

COMPARATIVE ANALYSIS

This chapter compares the performances of different classification schemes and fusion techniques used throughout this thesis for target differentiation and localization of commonly encountered features in indoor robot environments. To the best of our knowledge, a compact, complete and neat comparison of these classification and fusion methods supported by experimental verification does not exist for target classification. One of the main contribution of this thesis is the comparison of these methods based on experimental data.

The target differentiation algorithm summarized in Section 2.2, Dempster-Shafer evidential reasoning, simple majority voting and majority voting with preference ordering and reliability measures are employed in five experimental test areas in Chapters 2 and 3. Fuzzy c-means clustering algorithm is applied to find the optimum number of classes existing in the training set in Chapter 4. In order to compare these techniques with the other classification schemes applied in this thesis in a common test pool, these techniques are also evaluated with test sets I, II, and III, described in Section 5.6 and used throughout this thesis.

Initially, the target differentiation algorithm is employed at each angular step to

method	% of correct classif.	% of correct r estimation				% of correct θ estimation			
		error tolerance ϵ_r				error tolerance ϵ_θ			
		±0.125 cm	±1 cm	± 5 cm	±10 cm	±0.25°	±2°	±10°	±20°
DA	61	16	36	72	80	19	41	59	95
D-S	89	17	36	72	80	32	61	98	98
SMV	82	16	36	72	80	19	41	61	98
rel_i=1	88	16	36	72	80	19	41	61	98
rel_i^1	90	29	48	82	89	32	61	98	98
rel_i^2	90	29	48	82	89	32	61	98	98
rel_i^3	92	17	36	72	80	20	44	67	97
rel_i^4	91	17	36	72	80	20	42	63	97
rel_i^5	94	16	36	72	80	19	41	59	96

Table 7.1: The percentages of correct classification, range (r) and azimuth (θ) estimation for differentiation algorithm (DA), Dempster-Shafer (D-S) fusion, simple majority voting (SMV), and majority voting schemes with different reliability measures for test set I.

method	% of correct classif.	% of correct r estimation				% of correct θ estimation			
		error tolerance ϵ_r				error tolerance ϵ_θ			
		±0.125 cm	±1 cm	± 5 cm	±10 cm	±0.25°	±2°	±10°	±20°
DA	57	16	35	60	77	19	41	60	95
D-S	85	20	39	61	77	31	56	98	98
SMV	79	16	36	60	77	19	42	62	98
rel_i=1	84	16	36	60	77	19	42	62	98
rel_i^1	86	28	47	83	85	31	56	98	98
rel_i^2	86	28	47	83	85	31	56	98	98
rel_i^3	88	20	40	61	77	23	48	78	97
rel_i^4	87	20	39	61	77	23	47	77	96
rel_i^5	91	16	35	60	77	19	41	60	96

Table 7.2: The percentages of correct classification, range (r) and azimuth (θ) estimation for differentiation algorithm (DA), Dempster-Shafer (D-S) fusion, simple majority voting (SMV), and majority voting schemes with different reliability measures for test set II.

determine the target type. After determining the target type, range and azimuth of the targets are also estimated [48]. Then, Dempster-Shafer evidential reasoning and various voting schemes are used to fuse decisions made at each of the 58 angular steps to reach a single decision for a pattern set. In these fusion techniques, measurements collected at

method	% of correct classif.	% of correct r estimation				% of correct θ estimation			
		error tolerance ϵ_r				error tolerance ϵ_θ			
		$\pm$0.125 cm	$\pm$1 cm	$\pm$ 5 cm	$\pm$10 cm	$\pm$0.25°	$\pm$2°	$\pm$10°	$\pm$20°
DA	61	16	35	62	74	19	40	56	86
D-S	87	16	35	62	74	26	54	99	99
SMV	80	16	35	62	74	19	40	56	86
rel_i=1	85	16	35	62	74	19	40	56	86
rel_i^1	88	23	43	72	81	26	54	99	99
rel_i^2	88	23	43	72	81	26	54	99	99
rel_i^3	91	16	35	62	74	19	40	56	86
rel_i^4	89	16	35	62	74	19	40	56	86
rel_i^5	92	16	35	62	74	19	40	56	86

Table 7.3: The percentages of correct classification, range (r) and azimuth (θ) estimation for differentiation algorithm (DA), Dempster-Shafer (D-S) fusion, simple majority voting (SMV), and majority voting schemes with different reliability measures for test set III.

each angular step are considered as the measurements taken from a single sensor node and 58 decisions are fused to reach a final decision on target type for a single pattern set. Moreover, weighted averages of the 58 range and azimuth estimates in a pattern set are calculated to find the range and the azimuth of the target. In Dempster-Shafer evidential reasoning, these weights are the ratio of the belief values assigned to range and azimuth estimates at each angular step to the sum of the belief values assigned to range and azimuth estimates at all 58 angular steps. In simple majority voting, these weights are taken as 1/58. In the different voting schemes including preference ordering and reliability measures, the ratio of reliability assigned at each angular step to the sum of the reliabilities assigned at all 58 angular steps are used as weights. In this case, preference orders are taken as the belief values assigned to each target type at each angular step. The resulting average percentages over all target types for correct type classification, correct range and correct azimuth estimation for test sets I, II, and III are given in Tables 7.1–7.3 respectively. A range or azimuth estimate is again considered correct if it is within an error tolerance of ϵ_r or ϵ_θ of the actual range or azimuth respectively. For all test sets, inclusion of preference orders over target types and assignment of

reliability measures in the fusion process always brings some improvement compared to the results of simple majority voting. The fifth reliability measure gives the highest percentage of correct differentiation, and is followed by the third, fourth, first, and second measures. These five reliability measures always result in better classification performance than the assignment of equal reliability measure to all sensor nodes. In addition, their performances are also better than that of Dempster-Shafer evidential reasoning. The highest percentage of correct classification is obtained with the test data I, and is followed by test data III ($0-2\%$ decrease) and II ($3-4\%$ decrease). For all test sets, azimuth estimation results are slightly better than range estimation results. Note that, these methods do not require any training, therefore training set is not used for these methods.

Test set I			Test set II			Test set III		
$\mathbf{X_I}$	$\mathbf{X_{II}}$	$\mathbf{X_{III}}$	$\mathbf{X_I}$	$\mathbf{X_{II}}$	$\mathbf{X_{III}}$	$\mathbf{X_I}$	$\mathbf{X_{II}}$	$\mathbf{X_{III}}$
98	93	93	94	92	92	91	92	92

Table 7.4: The percentages of correct classification for three test sets with three vector representations obtained by employing the fuzzy c-means clustering algorithm.

Next, the fuzzy c-means clustering algorithm is employed to find the cluster centers with total number of clusters $c = 7$ for three different design sets extracted from the training set used throughout this thesis using three different vector representations $\mathbf{X_I}$, $\mathbf{X_{II}}$, and $\mathbf{X_{III}}$ introduced in Section 4.3. Then, these cluster centers are used to classify the vector representations of target types in each test set. The resulting average percentages of correct classification over all target types for each test data with corresponding vector representations are given in Table 7.4. Referring to Table 7.4, the highest percentage of correct classification is obtained with test set I, followed by test set II and III. Moreover, the results of test set I and II are comparable. For all test sets except test set III, vector representation $\mathbf{X_I}$ gives better results and vector representations $\mathbf{X_{II}}$ and $\mathbf{X_{III}}$ result in equal average percentage of correct classification.

method	targets discriminated	differentiation accuracy (%)	r/θ est (%)	training data	learning	parametric
target differentiation algorithm [21,22]	P,C,AC	61	yes 16/19	not used	no	no
Dempster-Shafer evidential reasoning [44]	P,C,AC, {E,CY,U}	89	yes 20/31	not used	no	no
simple majority voting [44]	P,C,AC, {E,CY,U}	82	yes 16/19	not used	no	no
voting with reliability measures [145]:	P,C,AC,		yes	not used	no	no
rel_i=1	{E,CY,U}	88	16/19			
$rel_i^1 = m(r_i)m(\theta_i)$		90	29/32			
$rel_i^2 = \min\{m(r_i), m(\theta_i)$		90	29/32			
$rel_i^3 = \frac{m(r_i)+m(\theta_i)}{2}$		92	20/23			
$rel_i^4 = \max\{m(r_i), m(\theta_i)$		91	20/23			
$rel_i^5 = m$(first choice) $- m$(second choice)		94	16/19			
statistical pattern recognition [146]:	P,C,AC,E,CY		no	used	no	
k-NN		98		stored		no
generalized k-NN		99		stored		no
kernel estimator		99		stored		no
PDE with:						
heteroscedastic NM		81		not stored		yes
homoscedastic NM		76		not stored		yes
linear discriminant analysis		71		not stored		no
fuzzy c-means clustering [147]	P,C,AC,E,CY	98	no	used not stored	yes	no
neural networks [148]:	P,C,AC,E,CY		yes(yes)[no]	used not stored	yes	no
raw signal		95 (95) [95]	79/89(73/95)			
DWT		98 (99) [97]	71/90(80/92)			
DFT		99 (98) [98]	63/91(72/94)			
DFRT		100 (98) [97]	75/93(68/86)			
DHT		99 (97) [97]	67/84(62/84)			
KSOFM		78 (76) [13]	24/69(21/66)			
KSOFM with linear classifier [149]	P,C,AC,E,CY	85	yes 42/80	used not stored	yes	no

Table 7.5: Overview of the methods compared. The target types enclosed in braces can be resolved only as a group. The numbers before the parentheses are for non-modular networks trained by the back-propagation algorithm and the numbers in parentheses are for modular networks, whereas the numbers in brackets are for networks trained with the generating-shrinking algorithm.

Up to this point, the results of target differentiation algorithm and fusion techniques based on this algorithm for target differentiation and localization with sonar are compared separately. The performance of different neural network structures trained with two different training algorithms with different input signal representations are also compared separately in Chapter 5. Moreover, the results of statistical pattern recognition techniques for target differentiation with sonar are compared among themselves in Chapter 6. In order to give an overview and to make a comparison of all differentiation

schemes and fusion techniques employed in this thesis, Table 7.5 is constructed. In this table, highest average percentages of correct classification, correct range and azimuth estimation for the error tolerances $\epsilon_r = 0.125$ cm and $\epsilon_\theta = 0.25°$ obtained with each method, independent of vector representations of patterns and test set, are given. Referring to this table, with the target differentiation algorithm and the fusion techniques based on this algorithm (Dempster-Shafer evidential reasoning and various voting schemes), only three of the target types employed in this thesis (plane, corner, and acute corner) can be differentiated. However, all target types can be differentiated with the other methods employed in this thesis. The fact that the other methods are able to distinguish all target types indicates that they must be making more effective use of the available data than the target differentiation algorithm. Statistical pattern recognition techniques, fuzzy c-means clustering algorithm, and neural networks trained with the generating-shrinking algorithm cannot be used for target localization unlike all of the other methods. Target differentiation algorithm and fusion techniques employed based on this algorithm do not require a training set, whereas all of the other methods do. Moreover, non-parametric statistical pattern recognition techniques store their training sets in the testing phase. The highest average percentages of correct classification is 100% and is obtained with non-modular neural network trained with back-propagation algorithm employing the fractional Fourier transform. Although this network structure gives the highest percentage of correct classification, the better localization of the targets is obtained with the modular neural network trained with back-propagation algorithm employing the wavelet transform with 99% average percentage of correct classification. The lowest percentage of correct classification is 61% and is obtained with the target differentiation algorithm.

For most cases, vector representation $\mathbf{X_I}$ gives the best results, followed by $\mathbf{X_{II}}$ and $\mathbf{X_{III}}$ in the given order. Note that different vector representations are not applicable to the target differentiation algorithm and fusion techniques employed based on this algorithm since they determine the target type by using differential signals $\mathbf{X_{II}}$ obtained

from the original signals $\mathbf{X_I}$ and they assign belief values to their decisions using $\mathbf{X_{III}}$. For most cases, the results obtained with test set I are better than those obtained for test set II and III. This is followed by test set II and test set III for all methods except the target differentiation algorithm and the fusion techniques employed based on this algorithm. However, the gap between the results of these test sets and test set I is higher for statistical pattern recognition techniques than that for all other methods.

In this chapter, a comparison of all approaches used for target classification and localization throughout this thesis is made in a common test pool experimentally. The results provided in this thesis are vital for robotics researchers who are looking for which method results in better target classification and localization performance with sonar. In the next chapter, an application of the best classification scheme to build map of mobile robot's environments using a new exploring strategy based on the generalized Voronoi diagram is presented.

Chapter 8

MAP-BUILDING WITH SONAR

So far, we have employed a number of classification and localization schemes to classify and localize commonly encountered target types in a mobile robot's environment. These methods are compared on the basis of their capability of classification and localization of these targets on different test data sets. In this chapter, application of the *best* classification scheme (which is the neural network classifier) resulting in the highest percentages in both target classification and localization in map-building with sonar is provided. A new exploring strategy based on the Generalized Voronoi Graph [45, 46] is used.

This chapter is organized as follows: definition of Generalized Voronoi Diagram and the meet points are provided in Section 8.1. In Section 8.2, the *best* classification scheme, which is the neural network classifier, is applied at the meet points to build the global map of a mobile robot's environment.

8.1 Generalized Voronoi Diagram

Generalized Voronoi diagram (GVD) is a technique used in motion planning of a mobile robot in [45–47, 150]. The basic underlying idea behind the usage of GVDs in robot's motion planning is that it provides maximal clearance between the robot and the obstacles with no *a priori* knowledge of the robot's environment. The simplest definition of the GVD is that it is the locus of points which are equidistant to two points in an m-dimensional work space. The locus of points which are equidistant to m points is referred as the *generalized Voronoi graph* (GVG). Before providing mathematical definitions of GVD and GVG, it is necessary to define a distance function and its gradient. To make this definition, assume that a point robot operates in a work space, $\mathcal{W}$, of an m-dimensional Euclidean space which consists of convex obstacles $C_1, \cdots, C_n$. In this work space, non-convex obstacles are considered as the union of convex shapes. The distance between a point x and an obstacle C_i is defined as the shortest Euclidean distance between the point and all points of the obstacle. Therefore, the distance function and its gradient are defined as

$$d_i(x) = \min_{c_\circ \in C_i} ||x - c_\circ|| \quad \text{and} \quad \nabla d_i(x) = \frac{x - c_\circ}{||x - c_\circ||} \tag{8.1}$$

The set of points equidistant to two obstacles C_i and C_j is the basic building block of the GVD and GVG which is called a *two-equidistant face*:

$$\mathcal{F}_{ij} = \{x \in \Re^m : \ 0 \leq d_i(x) = d_j(x) \leq d_h(x) \ \forall h \neq i, j \text{ and } \nabla d_i(x) \neq \nabla d_j(x)\} \tag{8.2}$$

The union of two-equidistant faces is the GVD. The intersection of two-equidistant faces $\mathcal{F}_{ij}$, $\mathcal{F}_{ik}$, with $\mathcal{F}_{jk}$ is called three-equidistant face represented by $\mathcal{F}_{ijk}$ (Figure 8.1). This intersection procedure is continued until m-equidistant faces $\mathcal{F}_{i_1 \ldots, i_m}$ are obtained. The GVG is the collection of m-equidistant faces and $(m+1)$-equidistant faces. The m-equidistant faces are called *generalized Voronoi edges* and $(m+1)$-equidistant faces are always points and are called *meet points*. Note that GVD and GVG coincide in the planar case. The most attractive advantage of GVD is that it can be constructed by using only

the easily available range information in an unknown environment. The incremental construction of GVD for a cylindrical robot by using ultrasonic range data is presented in [46].

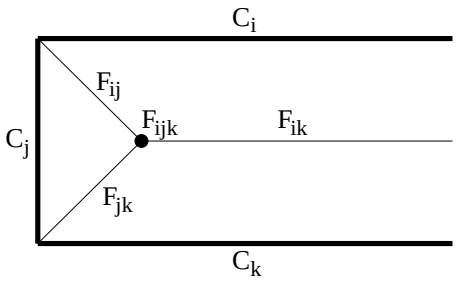


Figure 8.1: A GVD example.

8.2 Experimental Studies

At the Bilkent University Robotics Research Laboratory, nine different test environments consisting of commonly encountered target types is constructed. The target types used to construct these environments are planes, corners, acute corner of $\theta_c = 60°$, edges of $\theta_e = 90°$ and $120°$, cylinders with radii $r_c = 2.5,\ 5$, and 7.5 cm. These rooms and the position of their meet points are represented in Figures 8.2 and 8.3. In this study, meet point definition is extended to include junctions of the corners and acute corners as a point target. This extended definition of the meet points is reasonable since the robots equipped with ultrasonic sensor configuration used in this thesis can measure distances to these junctions. In Figure 8.4, the GVD of room 2 and 3 are represented. In this figure, regular meet points are plotted as solid dots, whereas extended meet points are plotted as empty circles. For example, meet point 4 in room 2 is an extended meet point which is equidistant to the corners A and F and the edge E. Meet points 6 and 7 in room 3 are also extended meet points. Meet point 6 is equidistant to the corners C and D and the edge H, and meet point 7 is equidistant to the corners A and B and the edge E. Similarly meet point 2 in both room 2 and 3 is also an extended meet point

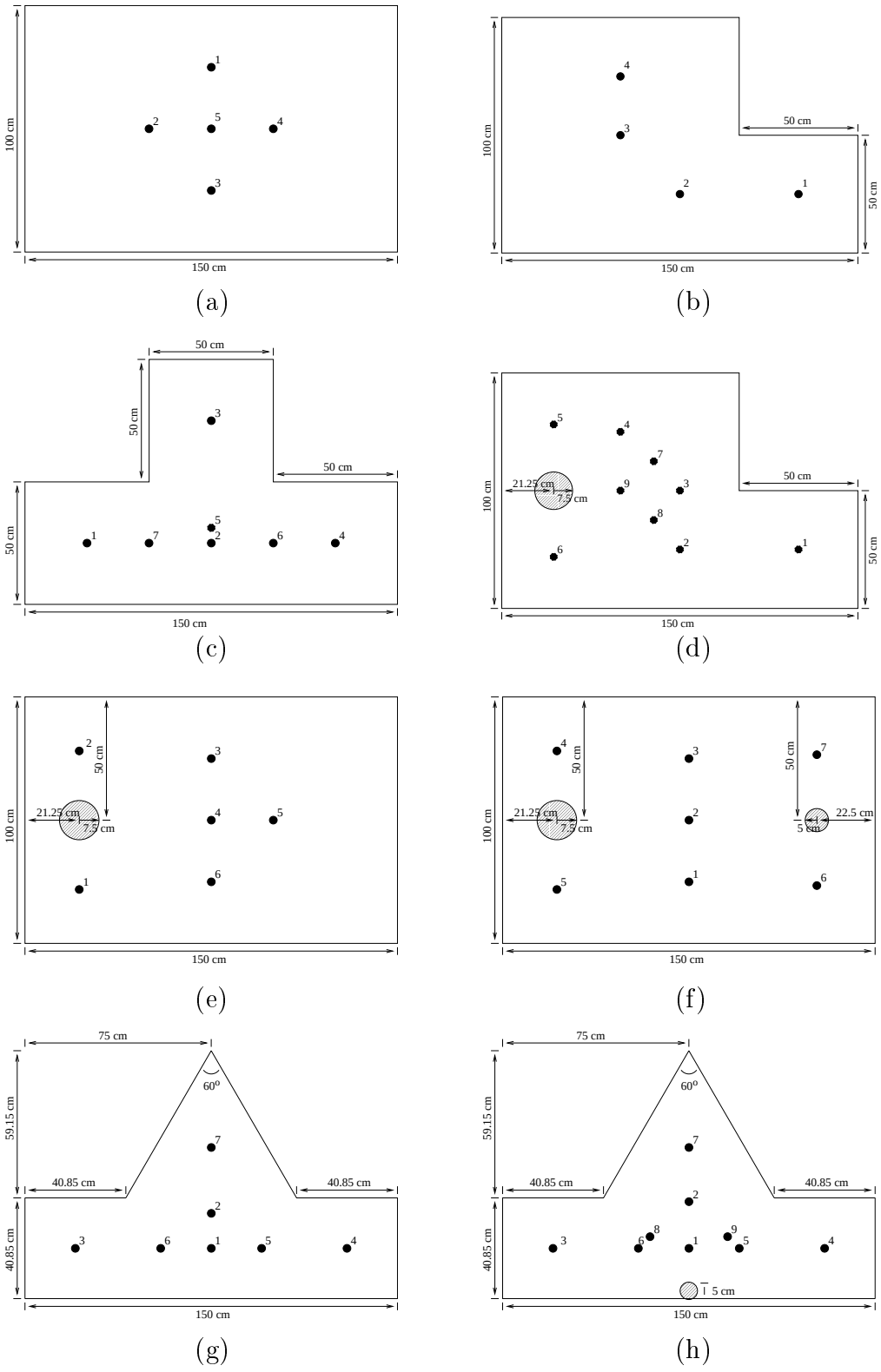


Figure 8.2: Experimental test environments and their meet points. Rooms (a) 1, (b) 2, (c) 3, (d) 4, (e) 5, (f) 6, (g) 7 (h) 8.

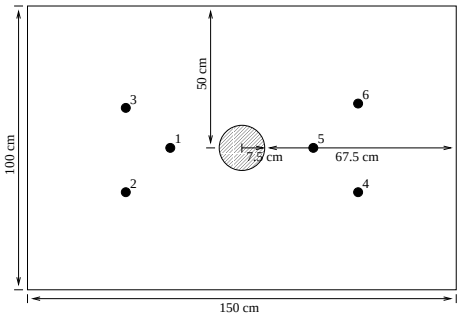
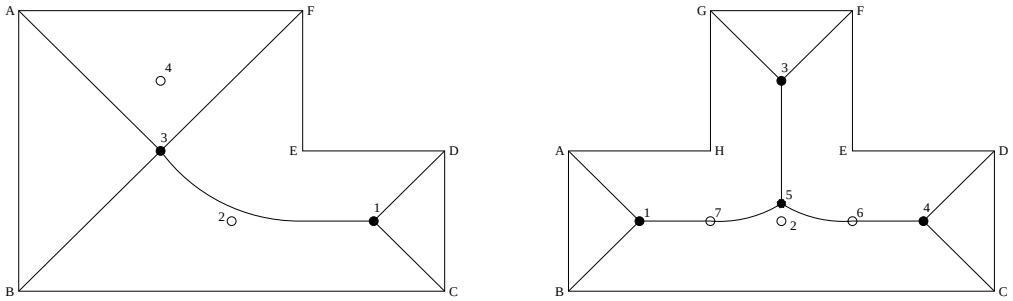


Figure 8.3: Experimental test room 9 and the meet points of this room.



(a) (b)

Figure 8.4: The GVD of (a) room 2 and (b) room 3.

which is equidistant to the corners B, C, and D of the corresponding rooms. These nine rooms are scanned with the sensor node (a pair of ultrasonic transducers with separation $d = 25$ cm mounted on a stepper motor) used throughout our studies from $\alpha = 1.8°$ to $360°$ with $1.8°$ increments at all of the meet points of the rooms involved. During this scanning procedure, 200 $(= \frac{360}{1.8°})$ angular samples of each of $A_{aa}(\alpha)$, $A_{bb}(\alpha)$, $A_{ab}(\alpha)$, $A_{ba}(\alpha)$, $t_{aa}(\alpha)$, $t_{bb}(\alpha)$, $t_{ab}(\alpha)$, $t_{ba}(\alpha)$ are acquired at each meet point position to be given to a neural network which is found to be the *best* classification scheme based on our previous studies.

Referring to the comparative analysis made in Chapter 7 [149], the highest correct classification rate of 100% is obtained with a non-modular neural network trained with the back-propagation algorithm employing the 0.25'th-order fractional Fourier

transform of input signal I_1. Although this network structure gives the best classification performance, best localization of the targets is obtained with the modular neural network trained with the back-propagation algorithm employing the low-frequency part of DWT of I_1 at the second level with a correct classification rate of 99%. For this reason, modular neural network classifier trained with back-propagation algorithm employing the low-frequency part of DWT of I_1 is taken as the best classifier. However, usage of this network to extract the map of these nine rooms limits the accuracy of the resulting maps from the beginning, since the networks considered in our previous work have only seven neurons at their output layers to encode the range and azimuth of the targets. With these seven neurons, only the range values from 30 cm to 61.75 cm with 0.25 cm increments, and the azimuth values from -30° to $+33.5^\circ$ with 0.5° increments can be encoded. However, the minimum range value to be encoded with this network in these nine rooms is 15.4 cm and the maximum range value is 150.9 cm, while the minimum azimuth value is -68.4° and the maximum azimuth value is $+68.4^\circ$. For this reason, re-training of this neural network classifier is inevitable at least for range and azimuth estimation. Input signals given to the previous networks are obtained by scanning each target separately at each training location from $\alpha = -52^\circ$ to 52° with 1.8° increments. This larger angular scan range is used to be able to see the targets with the sensor node from all training positions in the work space used in our previous studies. However, in the test rooms, usually more than one target is seen with this large angular scan range. It is also better to re-train these networks for target identification and localization with smaller angular scan range.

In order to re-train this modular neural network with the low-frequency part of DWT of I_1 at second level, a training set is constructed by scanning these nine rooms at their meet points from $\alpha = 1.8^\circ$ to 360° with 1.8° increments. Three similar sets of scans are collected for each room at each meet point, resulting in $36,000 (= 3$ data sets $\times$ 60 meet points $\times$ 200 angular samples of each signal at each meet point) sets of signals to be used for training.

At each scan angle, 16 angular samples of each signal are used to classify and localize the target in front of the sensor node. These samples are chosen at each scan angle α such that the first eight of these samples correspond to the samples at the previous eight scan angles, the ninth sample is the sample at the current scan angle, and the last seven samples correspond to the samples of the succeeding seven scan angles. Each module has an equal number of input-layer neurons which is equal to 24 ($= \frac{1}{2^j} \times$ 16 angular samples $\times$ 6 signals for I_1, and $j = 2$ for the second level of the DWT). We have also considered 8 angular samples of each signal, but since 16 samples resulted in better classification accuracy, we have used 16 angular samples of each signal for classification and localization of the targets. The number of hidden-layer neurons is equal to 20, 50, and 45 for the modules used to identify target type and estimate its range and azimuth respectively, which are determined by *enlarging* as before. The number of output-layer neurons of the modules for target identification, range and azimuth estimation is 7, 10 and 9, respectively. The target range r is binary coded with a resolution of 0.25 cm with ten output neurons. Therefore, a range value from 0 to 255.75 cm can be coded in this case. The azimuth θ of the target with respect to the line-of-sight of the sensing node is also binary coded with resolution 0.5° with nine output neurons. Therefore, a range value from -128.4° to $+127.1^\circ$ can be coded in this case.

After training this modular network with the back-propagation algorithm, this network is used to extract the map of the nine rooms introduced before with the signals acquired at each meet point which are not used in the training phase. For comparison, we have also provided the results obtained by using the previous network which is the network using 58 angular samples of each signal obtained in our previous research by training the network with 700 sets of signals taken at 25 different training locations from seven different target types. This network also uses the low-frequency component of the DWT of the signal I_1 at the second level ($j = 2$).

The feature-based maps of the nine rooms considered in this study are extracted at

room / meet point	1	2	3	4	5	6	7	8	9
1	95(86)	81(55)	74(53)	85(55)	82(67)	70(56)	63(39)	31(34)	86(57)
2	93(78)	81(71)	60(52)	73(69)	84(64)	88(77)	41(29)	39(40)	81(56)
3	83(68)	90(85)	68(39)	88(69)	81(70)	53(46)	82(47)	74(35)	75(55)
4	91(69)	86(83)	79(51)	62(50)	89(76)	85(57)	80(46)	74(31)	83(58)
5	93(85)	–	74(44)	83(59)	88(62)	85(62)	60(44)	54(34)	83(66)
6	–	–	67(60)	73(59)	77(66)	80(55)	62(47)	61(33)	76(60)
7	–	–	65(64)	65(56)	–	85(58)	61(50)	57(35)	–
8	–	–	–	77(49)	–	–	–	50(41)	–
9	–	–	–	78(66)	–	–	–	57(36)	–

Table 8.1: Percentages of correct classification at each meet point of all nine rooms.

their meet points by specifying the target types at each viewing angle by the newly-trained neural network. These maps are presented in Figures 8.5–8.18. In all of these figures, $*$ represents meet point position, $\square$, $+$, $\star$, $\times$, $\triangle$, $\diamond$, and $\circ$ correspond to plane, corner, acute corner, edge, and cylinders with radii $r_c = 2.5$, 5.0, and 7.5 cm, respectively. Moreover, the percentages of correct classification at these meet points are also calculated and tabulated in Table 8.1. In this table, the numbers given in parentheses correspond to the results obtained by employing the previous neural network classifier. Referring to this table, highest percentages of correct classification in each room are 95%, 90%, 79%, 88%, 89%, 88%, 82%, 74%, and 86% obtained at the 1, 3, 4, 3, 4, 2, 3, 3, and 1st meet points of the corresponding rooms. Similarly, the lowest percentages of correct classification in each room are 83%, 81%, 60%, 62%, 77%, 53%, 41%, 31%, and 75% obtained at the 3, 1 and 2, 2, 4, 6, 3, 2, 1, and 3rd meet points of the corresponding rooms. The percentages of correct classification obtained with our previously-trained network are most of the time lower than the corresponding percentages obtained with the newly-trained network. The average decrease in the percentages of correct classification is around 18%. With the previous network, highest percentage of classification among all rooms is obtained at the first meet point of the first room as 86% and the lowest one

is obtained at the fourth meet point of the 8th room as 31%.

room / meet point	1	2	3	4	5	6	7	8	9
1	4.0(5.0)	4.2(6.0)	4.5(6.3)	3.4(4.9)	5.9(7.4)	4.6(5.2)	5.5(4.9)	6.9(5.8)	5.0(6.1)
2	2.8(2.9)	4.1(5.1)	5.2(6.8)	4.3(5.2)	6.5(8.1)	4.2(3.8)	4.6(4.1)	7.1(5.6)	4.1(4.8)
3	3.8(4.3)	4.1(4.0)	5.4(8.2)	6.2(6.8)	4.0(3.7)	12.7(10.1)	4.6(6.6)	5.5(8.2)	3.8(3.9)
4	4.2(5.1)	3.3(3.8)	3.9(6.0)	3.8(4.5)	3.8(3.5)	6.5(8.8)	4.6(6.6)	5.2(6.7)	5.0(6.9)
5	3.1(3.5)	–	5.1(7.0)	6.1(8.0)	4.0(4.1)	4.5(5.7)	4.5(4.3)	5.1(4.2)	5.3(6.7)
6	–	–	6.7(7.0)	7.4(8.4)	5.3(5.8)	8.5(9.7)	5.6(5.5)	5.6(4.8)	7.0(8.0)
7	–	–	6.3(7.5)	6.6(6.9)	–	6.0(6.3)	5.5(5.5)	6.5(6.8)	–
8	–	–	–	6.9(6.8)	–	–	–	5.3(4.3)	–
9	–	–	–	4.9(4.4)	–	–	–	4.8(4.3)	–

Table 8.2: Means and standard deviations of absolute range errors at each meet point of all nine rooms obtained by employing newly-trained neural network classifier.

room / meet point	1	2	3	4	5	6	7	8	9
1	4.5(3.8)	4.8(4.2)	4.6(4.2)	4.6(4.3)	5.1(4.5)	4.1(3.8)	5.0(4.5)	10.2(9.3)	5.8(4.6)
2	4.2(3.9)	4.6(3.5)	4.3(3.8)	4.1(3.8)	5.5(4.4)	5.2(3.9)	6.9(5.7)	8.3(7.8)	4.9(5.1)
3	4.5(3.9)	4.4(3.6)	4.7(4.2)	5.4(4.6)	4.6(4.1)	7.1(5.6)	5.8(4.3)	7.9(8.2)	4.5(4.6)
4	4.2(3.4)	4.5(3.9)	4.5(4.0)	4.4(3.9)	4.8(3.9)	5.5(4.8)	5.4(4.2)	6.6(7.0)	6.0(5.1)
5	4.6(3.6)	–	4.2(3.5)	5.3(4.2)	4.2(3.6)	4.4(3.9)	5.6(4.5)	5.8(4.5)	6.1(5.1)
6	–	–	4.8(4.4)	5.3(4.3)	4.2(3.8)	5.6(5.2)	5.8(4.6)	5.7(4.6)	6.8(6.0)
7	–	–	5.9(5.0)	4.8(3.9)	–	4.5(4.0)	5.5(4.8)	7.5(6.6)	–
8	–	–	–	4.4(3.7)	–	–	–	5.3(4.2)	–
9	–	–	–	5.2(3.6)	–	–	–	5.3(4.3)	–

Table 8.3: Means and standard deviations of absolute azimuth errors at each meet point of all nine rooms obtained by employing newly-trained neural network classifier.

In Tables 8.2 and 8.3, range and azimuth estimation errors at each meet point of each of the nine rooms are tabulated. Corresponding results for our previous network are given in Tables 8.4 and 8.5. In these tables, the numbers before the parentheses represent the means of the absolute range and azimuth errors, whereas the numbers given in parentheses correspond to the standard deviations of the absolute range and azimuth errors in centimeters and degrees, respectively. The range and azimuth errors obtained with the previous network are much larger than those obtained with the newly-trained

network, as expected. For the newly-trained network, means and standard deviations of the absolute range error are in the range of 2.8−12.7 and 2.9−10.1 cm, respectively (Table 8.2). The means and standard deviations of the absolute azimuth error are in the range of 4.1°−10.2° and 3.4°−9.3° (Table 8.3). For the previously-trained network, means and standard deviations of the absolute range error obtained are between 10.3−45.8 and 7.5−37.6 cm (Table 8.4), whereas those of the absolute azimuth error are in the range of $20.3° - 28.4°$ and $12.7° - 19.1°$, respectively (Table 8.5).

room / meet point	1	2	3	4	5	6	7	8	9
1	30.1(14.5)	27.8(30.8)	23.5(27.7)	27.1(31.7)	45.8(36.5)	28.8(16.1)	22.8(10.4)	26.2(11.6)	22.5(23.3)
2	23.6(22.6)	26.4(11.0)	23.1(7.5)	25.8(13.2)	42.4(36.7)	24.2(16.1)	21.9(6.7)	26.7(9.4)	23.8(24.3)
3	30.0(14.3)	15.5(19.2)	10.3(7.6)	23.1(14.4)	27.4(12.9)	29.9(14.9)	26.8(27.0)	20.2(18.9)	23.4(21.6)
4	25.5(23.3)	20.5(20.4)	23.8(27.2)	19.9(19.6)	22.8(14.4)	42.9(37.6)	25.8(25.5)	21.4(20.6)	22.0(23.2)
5	22.0(13.2)	–	21.2(8.3)	33.5(31.4)	23.8(24.2)	44.4(35.4)	23.6(13.9)	22.7(15.0)	23.9(25.5)
6	–	–	21.1(18.0)	38.4(28.0)	29.4(14.8)	41.3(34.7)	25.1(16.3)	24.0(14.2)	24.9(22.4)
7	–	–	26.1(21.2)	19.4(17.9)	–	43.2(36.9)	17.1(14.9)	21.4(16.0)	–
8	–	–	–	22.0(15.4)	–	–	–	23.0(16.0)	–
9	–	–	–	21.3(18.5)	–	–	–	22.8(14.4)	–

Table 8.4: Means and standard deviations of absolute range errors at each meet point of each of the nine rooms obtained by employing our previous neural network classifier.

room / meet point	1	2	3	4	5	6	7	8	9
1	23.3(15.0)	21.1(15.6)	21.3(14.5)	21.8(14.6)	25.6(17.1)	24.0(13.0)	22.6(15.9)	28.4(17.7)	21.4(15.7)
2	22.8(13.9)	25.2(13.6)	22.3(13.2)	24.7(13.0)	23.6(16.0)	23.1(15.3)	24.3(18.1)	26.0(18.3)	23.4(13.4)
3	23.9(13.9)	20.5(14.2)	22.2(13.7)	20.6(14.1)	24.4(13.5)	22.4(15.0)	23.2(13.0)	25.9(16.4)	25.2(16.9)
4	21.3(13.9)	20.3(14.2)	21.9(14.0)	21.8(13.9)	23.2(15.8)	24.0(16.1)	23.5(14.4)	22.3(13.9)	27.4(19.1)
5	24.1(15.2)	–	22.3(13.2)	22.3(15.0)	21.9(13.6)	23.6(15.8)	25.3(14.0)	23.5(12.9)	22.5(14.7)
6	–	–	23.1(13.8)	23.9(14.8)	23.1(12.9)	23.7(15.3)	23.1(14.1)	23.3(14.7)	23.4(14.4)
7	–	–	21.4(13.1)	20.5(15.1)	–	23.5(14.7)	22.0(14.1)	24.4(16.8)	–
8	–	–	–	24.6(12.7)	–	–	–	21.0(13.0)	–
9	–	–	–	23.0(13.9)	–	–	–	22.3(15.6)	–

Table 8.5: Means and standard deviations of absolute azimuth errors at each meet point of each of the nine rooms obtained by employing our previous neural network classifier.

After extracting the *local* maps of all nine rooms at their meet points, fusion of these maps are considered to obtain a *global* map of each room. The centroid of the meet points of each room is found by averaging the x and y coordinates of all of the meet points in that room when the lower-left corner of the room is considered as the origin. For this fusion process, our previously employed fusion schemes which are Dempster-Shafer evidential reasoning, simple majority voting, and voting with preference ordering

and reliability measures are used. While using these fusion schemes, identification module outputs of the neural network classifiers are used to make basic probability mass assignments to the targets. However, it is not possible to take these values as they are since these values must be normalized to one for at least Dempster-Shafer evidential reasoning. After this normalization process, basic probability mass to an unknown target is assigned as $m_i(u) = 1 - m_i(\text{first choice})$ where $m_i(\text{target type})$'s are probability masses assigned to each target at meet point i by the neural network classifier (i.e., normalized output of each output neuron of the module used for target identification in the modular neural network classifier). Here, the basic probability mass assigned to unknown target represents ignorance by reflecting the possibility of being any target type including the classifier's first choice. Then, the basic probability mass values assigned to all targets including unknown are rescaled to add up to one. Since the seven target types employed in this study are disjoint, these basic probability masses are also equal to belief values assigned to each of these targets. In order to fuse the estimated range and azimuth values of a target at each viewing angle by using the neural network classifier at the meet points of each room, these range and azimuth values are projected to the centroid of the meet points of each room by using geometry. The fused range and azimuth values are found by taking weighted averages of these projected range and azimuth estimates at each viewing angle. The weights are chosen as the ratio of $m_i(\text{first choice})$ at the ith meet point to the sum of $m(\text{first choice})$s of all meet points of the corresponding room [i.e., $\sum_i m_i(\text{first choice})$] for Dempster-Shafer evidential reasoning. For simple majority voting, they are equal to 1/(total number of meet points in the corresponding room). For voting with preference ordering and reliability measures, they are equal to the ratio of the product of reliability measure assigned to the classifier and $m_i(\text{first choice})$ at meet point i to the sum of the products of reliability measures and $m(\text{first choice})$s of all meet points [i.e., $\sum_i rel_i^5 m_i(\text{first choice})$] in the corresponding room. Note that preference orders are taken again as the belief values assigned to each target type for voting with preference ordering and reliability measures. Although all five reliability

measures introduced in our previous studies are employed in this study, the fifth one [$rel_i^5 = m_i$(first choice) $- m_i$(second choice)] gives the best results for all nine rooms. For this reason, the results of the fifth reliability measure is provided throughout this thesis for the results of voting fusion with preference ordering and reliability measures.

The global maps obtained by using these three fusion schemes in all nine rooms are given in Figures 8.19−8.27. These maps are drawn with respect to the centroid of the meet points in each room. Referring to Figure 8.27, it can be seen that cylinder information has been lost by fusing decision of all classifiers with respect to the centroid of room 9 which is the center of the cylinder in the middle. In order to avoid this information loss, we consider two centroids for room 9 one of which corresponding to the centroid of the meet points existing on the left-hand side of this room, the other one corresponding to the centroid of the meet points existing on the right-hand side of this room. For the map extracted at each centroid of room 9, the decision of the classifiers at the corresponding 3 meet points are fused. The maps obtained at these two centroids in room 9 are given in Figures 8.28−8.29.

The percentages of correct classification obtained by employing these three fusion processes with respect to the centroid of each room are provided in Table 8.6. In this table, k-VRM stands for voting fusion with preference ordering and reliability measures in which classifiers are ordered on the basis of some selected criteria which will be explained later in this chapter. The comments on the results of this fusion scheme will be made later. In this table, the numbers given in parentheses correspond to the results obtained by employing the previously-trained neural network classifier. In Table 8.6, the percentages of correct classification for room 9 with respect to its original centroid (which is same as the center of the cylinder) are calculated by considering only the boundaries of this room since the cylinder information has been lost. If both the boundaries of room 9 and the borders of the cylinder existing in this room are considered, then the percentages of correct classification for room 9 with respect to its original centroid in Table 8.6 must

fusion scheme / room	D-S	SMV	VRM	k-VRM
1	95(82)	97(84)	98(88)	98(88)
2	92(80)	79(75)	95(86)	96(86)
3	84(72)	65(65)	86(75)	92(79)
4	90(76)	79(69)	91(79)	93(79)
5	91(79)	85(77)	94(84)	96(87)
6	87(75)	83(68)	92(79)	96(82)
7	68(57)	48(42)	75(62)	81(63)
8	61(46)	37(31)	68(54)	73(55)
9	90(72)	90(73)	94(77)	94(77)
9(1)	91(72)	84(57)	93(73)	93(73)
9(2)	89(69)	79(55)	92(73)	92(73)

Table 8.6: Percentages of correct classification obtained in each room with respect to their centroids by employing all four fusion schemes. Room 9(1) and 9(2) represent the left-hand side and right-hand side of room 9, respectively.

be halved. Referring to Table 8.6, the highest percentages of correct classification are obtained with voting fusion with preference ordering and reliability measures (VRM) for all rooms, and are followed by percentages of correct classification obtained with Dempster-Shafer evidential reasoning (D-S) and simple majority voting (SMV) for all rooms except the first one. These observations are also valid for the results obtained by employing the previous network. The maximum percentages of correct classification are 98% and 88% for the newly-trained network and the previous network respectively and are obtained in the first room, whereas the minimum percentages of correct classification are 68% and 54% for the same networks and are obtained in room 8. In Tables 8.7 and 8.8, absolute range and azimuth estimation errors for all fusion processes and all nine rooms are tabulated. In these tables, the numbers before the parentheses represent the means of the absolute range and azimuth errors, whereas the numbers given in parentheses correspond to the standard deviations of the absolute range and azimuth errors in centimeters and degrees, respectively. The means and standard deviations of absolute range and azimuth errors are comparable for all fusion schemes for both the

newly-trained network and the previous network. The range and azimuth errors with the our previous network are again higher than those obtained with the newly-trained network.

	range error (cm) (new network)				range error (cm) (previous network)			
method / room	D-S	SMV	VRM	k-VRM	D-S	SMV	VRM	k-VRM
1	1.6(1.4)	1.6(1.4)	1.5(1.4)	1.5(1.4)	21.2(13.3)	21.6(13.7)	20.8(11.8)	20.8(11.8)
2	3.6(3.4)	3.7(3.3)	3.3(3.4)	3.3(3.4)	20.2(11.6)	20.5(12.5)	18.9(12.0)	18.9(12.0)
3	3.9(3.2)	3.8(3.1)	4.0(3.2)	3.8(3.1)	21.2(6.8)	21.9(5.9)	20.2(8.3)	21.7(7.6)
4	3.9(2.9)	3.7(3.0)	3.8(3.2)	4.2(3.6)	18.9(12.7)	20.4(13.5)	17.5(13.0)	17.5(13.0)
5	3.1(2.7)	2.9(2.7)	3.0(2.6)	2.7(2.7)	23.8(14.5)	23.7(12.8)	22.5(11.2)	27.0(15.8)
6	3.7(2.5)	3.9(2.5)	3.8(2.6)	3.6(2.9)	23.2(8.3)	23.1(9.5)	23.0(7.4)	26.5(13.1)
7	3.5(1.9)	3.7(1.8)	3.7(2.3)	3.4(1.8)	21.9(5.6)	22.3(4.7)	21.1(7.3)	22.1(10.0)
8	4.8(2.6)	4.9(2.5)	4.7(3.0)	5.0(3.4)	24.9(7.2)	22.8(5.0)	23.6(7.6)	27.1(11.1)
9	6.2(3.2)	6.7(3.2)	5.3(3.1)	5.3(3.1)	24.2(10.8)	22.9(8.5)	20.8(7.3)	20.8(7.3)
9(1)	4.0(5.4)	4.5(5.7)	3.7(5.9)	3.7(5.9)	24.5(25.7)	24.4(26.3)	24.6(21.8)	24.6(21.8)
9(2)	4.4(5.3)	4.6(5.3)	3.9(5.2)	3.9(5.2)	25.2(27.7)	25.5(27.9)	25.1(27.3)	25.1(27.3)

Table 8.7: Means and standard deviations of absolute range errors with respect to the centroids of all rooms obtained by employing all four fusion schemes.

	azimuth error (deg) (new network)				azimuth error (deg) (previous network)			
method / room	D-S	SMV	VRM	k-VRM	D-S	SMV	VRM	k-VRM
1	2.2(2.4)	2.2(2.4)	2.1(2.3)	2.1(2.3)	20.3(12.9)	20.1(13.0)	20.2(12.5)	20.2(12.5)
2	3.0(2.9)	2.9(2.9)	2.8(2.9)	3.1(2.7)	19.5(12.9)	19.7(12.6)	19.2(12.8)	19.2(12.8)
3	2.5(2.3)	2.5(2.3)	2.5(2.4)	2.7(2.3)	19.4(10.8)	19.6(10.9)	18.7(11.1)	18.1(11.5)
4	3.0(2.8)	3.0(2.7)	2.9(2.9)	2.9(3.1)	18.2(11.9)	18.3(11.6)	18.0(11.5)	18.0(11.5)
5	3.0(2.9)	3.1(2.9)	2.8(3.0)	2.7(2.6)	18.8(13.0)	19.0(12.8)	19.2(12.6)	19.8(12.2)
6	3.9(3.1)	4.0(3.0)	3.7(3.3)	3.6(3.0)	19.3(12.4)	18.8(12.7)	20.0(11.3)	19.2(12.7)
7	3.2(3.1)	3.2(3.0)	3.2(3.1)	3.1(3.1)	22.0(13.7)	21.9(13.4)	22.7(13.3)	23.8(13.5)
8	5.6(5.8)	5.7(5.6)	5.6(6.3)	6.3(6.3)	24.8(16.8)	26.2(16.9)	25.3(16.4)	25.7(17.3)
9	5.3(4.2)	5.6(4.2)	4.7(4.3)	4.7(4.3)	19.5(12.5)	19.9(12.4)	19.6(12.9)	19.6(12.9)
9(1)	2.5(2.8)	2.6(3.0)	2.1(2.6)	2.1(2.6)	19.7(13.2)	19.8(13.2)	20.1(12.4)	20.1(12.4)
9(2)	3.0(3.2)	3.1(3.3)	2.8(3.0)	2.8(3.0)	21.0(13.2)	20.9(12.6)	21.3(13.2)	21.3(13.2)

Table 8.8: Means and standard deviations of absolute azimuth errors with respect to the centroids of all rooms obtained by employing all four fusion schemes.

For further improvement of correct classification performances, three different criteria

have been considered to order the classifiers employed at each meet point in the voting fusion with preference ordering and reliability measures which gives the best fusion results among the three fusion schemes. The first criterion used to order the classifiers employed at each meet point of a given room is the smallest range estimation. Starting with the classifier with the smallest range estimation to a certain reflection point, classifiers are added to the fusion list in increasing measured range to the same reflection point. Second criterion used to order the classifiers employed at each meet point of a given room is the smallest absolute azimuth estimation. Starting with the classifier with the smallest absolute value of azimuth estimate, classifiers are added to the fusion list in increasing order of absolute azimuth estimates. Finally, classifiers employed at each meet point of a given room are ordered based on their level of belief, irrespective of target type. Starting with the classifier with the highest belief, classifiers are added to the fusion list in the order of decreasing belief (highest belief). The objective for all three criteria is to select more informative classifiers such that if the target is closer to the surface of transducer (smallest range) and closer to the line-of-sight of the transducer (smallest azimuth), it is possible to classify this target more correctly. Similarly, the higher the belief of the first choice, the more correct is the target classification.

The results of the ordered voting fusion with preference ordering and reliability measures based on smallest range, smallest azimuth, and highest belief criteria for all rooms are given in Tables 8.9, 8.10, and 8.11, respectively. In all of these tables, k corresponds to the number of meet points (i.e., classifiers) employed in the fusion process and the numbers given in parentheses are the results obtained with the previous network. Maximum percentages of correct classification are obtained as 96%, 92%, 96%, 96%, and 73% with the smallest azimuth criteria for k = 3, 5, 5, 5, and 4 in rooms 2, 3, 5, 6, and 8, respectively. The maximum percentages of correct classification 93% and 81% are obtained with the highest belief criteria for $k = 3$ in the fourth room and with the smallest range criteria for k = 6 in room 7, respectively. The maximum percentages of correct classification are the same for all three criteria in rooms 1 and 9 which are

obtained by fusing the decisions of all classifiers and are equal to the results obtained by voting fusion with with preference ordering and reliability measures without any ordering.

room / k	1	2	3	4	5	6	7	8	9	9(1)	9(2)
1	84(66)	68(49)	42(32)	40(54)	58(57)	51(53)	48(37)	23(22)	47(30)	69(39)	57(51)
2	90(78)	80(67)	57(52)	58(58)	73(68)	66(60)	63(47)	44(36)	56(49)	84(55)	77(62)
3	95(88)	91(85)	73(72)	75(61)	88(78)	86(76)	80(51)	65(50)	66(56)	93(73)	92(73)
4	96(88)	95(86)	75(75)	83(68)	90(86)	87(78)	76(51)	66(51)	85(74)	–	–
5	98(88)	–	79(77)	86(70)	92(87)	92(78)	78(53)	66(54)	89(77)	–	–
6	–	–	82(79)	91(71)	94(84)	92(82)	81(61)	66(54)	94(77)	–	–
7	–	–	86(75)	91(75)	–	92(79)	75(62)	69(55)	–	–	–
8	–	–	–	92(77)	–	–	–	71(54)	–	–	–
9	–	–	–	91(79)	–	–	–	68(54)	–	–	–

Table 8.9: Percentages of correct classification obtained in each room with respect to the centroid of their meet points by employing ordered voting fusion with preference ordering and reliability measures for various k values when the classifiers are ordered by smallest range criteria.

room / k	1	2	3	4	5	6	7	8	9	9(1)	9(2)
1	84(76)	69(66)	51(42)	58(57)	71(62)	71(61)	46(37)	40(36)	73(52)	82(57)	70(52)
2	89(80)	82(74)	67(58)	70(67)	83(71)	81(70)	61(44)	52(42)	78(68)	85(61)	81(61)
3	95(87)	96(86)	82(74)	82(72)	89(82)	92(80)	77(52)	67(48)	87(73)	93(73)	92(73)
4	95(88)	95(86)	83(77)	88(75)	95(83)	94(80)	77(55)	73(50)	89(75)	–	–
5	98(88)	–	92(76)	88(76)	96(83)	96(79)	75(55)	73(51)	90(77)	–	–
6	–	–	88(76)	90(76)	94(84)	95(77)	75(60)	71(52)	94(77)	–	–
7	–	–	86(75)	91(79)	–	92(79)	75(62)	69(53)	–	–	–
8	–	–	–	90(79)	–	–	–	68(53)	–	–	–
9	–	–	–	91(79)	–	–	–	68(54)	–	–	–

Table 8.10: Percentages of correct classification obtained in each room with respect to the centroid of their meet points by employing ordered voting fusion with preference ordering and reliability measures for various k values when the classifiers are ordered by smallest azimuth criteria.

For our previous network, the maximum percentages of correct classification are obtained as 79%, 87%, 82%, and 55% with the smallest range criteria for $k = 6, 5, 6,$

room / k	1	2	3	4	5	6	7	8	9	9(1)	9(2)
1	92(78)	77(70)	72(55)	77(69)	85(70)	73(69)	55(46)	45(43)	88(65)	86(62)	87(67)
2	94(83)	85(77)	79(61)	85(72)	89(78)	81(73)	66(51)	56(50)	88(65)	90(67)	89(70)
3	96(87)	94(85)	85(76)	93(76)	94(84)	89(76)	75(60)	68(53)	93(71)	93(73)	92(73)
4	97(87)	95(86)	87(75)	91(77)	94(85)	91(78)	75(63)	68(54)	93(71)	–	–
5	98(88)	–	86(75)	91(78)	94(83)	92(79)	75(62)	68(54)	94(77)	–	–
6	–	–	86(75)	91(78)	94(84)	92(79)	75(61)	68(54)	94(77)	–	–
7	–	–	86(75)	91(79)	–	92(79)	75(62)	68(54)	–	–	–
8	–	–	–	91(79)	–	–	–	67(54)	–	–	–
9	–	–	–	91(79)	–	–	–	68(54)	–	–	–

Table 8.11: Percentages of correct classification obtained in each room with respect to the centroid of their meet points by employing ordered voting fusion with preference ordering and reliability measures for various k values when the classifiers are ordered by highest belief criteria.

and 7 in rooms 3, 5, 6, and 8, respectively. Maximum percentage of correct classification 63% is obtained with the highest belief criteria for $k = 4$ in room 7. For rooms 1, 2, 4, and 9, maximum percentages of correct classification are the same for all three criteria which are obtained by fusing the decisions of all classifiers and are equal to the results obtained by voting fusion with preference ordering and reliability measures without any ordering.

For both the newly-trained network and our previous network, these maximum percentages of correct classification are taken as the results of the voting fusion with preference ordering and reliability measures in which classifiers are ordered based on three criteria (k-VRM) and tabulated in the last column of Table 8.6, and resulting absolute range and azimuth errors are also tabulated in the columns labeled as k-VRM in Tables 8.7 and 8.8, respectively. Resulting global maps in all nine rooms with this fusion scheme by employing the newly-trained network are presented in Figures 8.19(d)−8.29(d).

During this study, a neural network classifier has been designed to identify the room being explored by the mobile robot. The inputs to this neural network are the x and y

coordinates of the meet points in a given room while considering the lower-left corner of each room as the origin. Therefore, the number of input-layer neurons of this network is 18 (2 neurons for each meet point and a maximum of 9 meet points). If the number of meet points in a given room is less than 9, then zeros are placed for x and y coordinates of the non-existing meet points in that room. This network has 9 output neurons, each one representing one of the nine rooms (i.e., the first output neuron corresponds to the first room, etc.). The number of hidden-layer neurons is found by enlarging as 10. A training set consisting of 100 input patterns is generated for each room resulting in 900 (100×9 rooms) training patterns by adding zero-mean Gaussian noise with a standard deviation of 3 cm to the correct meet point positions. After training is completed, this network is tested with a test data set consisting of 900 input patterns generated by adding zero-mean Gaussian noise with different standard deviation σ to the correct meet point positions. These σ values used for testing are 1.5, 3.0, 4.5, 6.0, 7.5, 9.0, 18.0, 27.0, and 30.0 cm. This network is also tested with the correct meet point positions (i.e., $\sigma = 0$). The percentages of correctly identifying the room being explored obtained for these noisy patterns are tabulated in Table 8.12. Referring to this table, a 100% success rate is maintained until $\sigma = 9.0$ cm. After this σ value, 98%, 91%, and 85% are obtained for σ values 18.0, 27.0, and 30.0 cm, respectively. It is seen that, if σ is increased ten times, the success rate decreases only by 15%. Therefore, this network can find applications in robotics where a mobile robot can find the meet points existing in a given environment from simple range measurements. Then, it can identify the environment which it is exploring with this method.

σ (cm)	0.0	1.5	3.0	4.5	6.0	7.5	9.0	18.0	27.0	30.0
%	100	100	100	100	100	100	100	98	91	85

Table 8.12: The percentages of correctly identifying the room which the mobile robot is exploring for various σ values.

In this chapter, application of the best classification scheme for building the map of a

mobile robot's environment is presented using a novel exploring strategy. A new neural network classifier is designed to classify different environments by using critical vantage points in these environments. In the next chapter, besides sonar, inclusion of physically different sensors such as infrared and structured-light systems is considered to improve correct target classification.

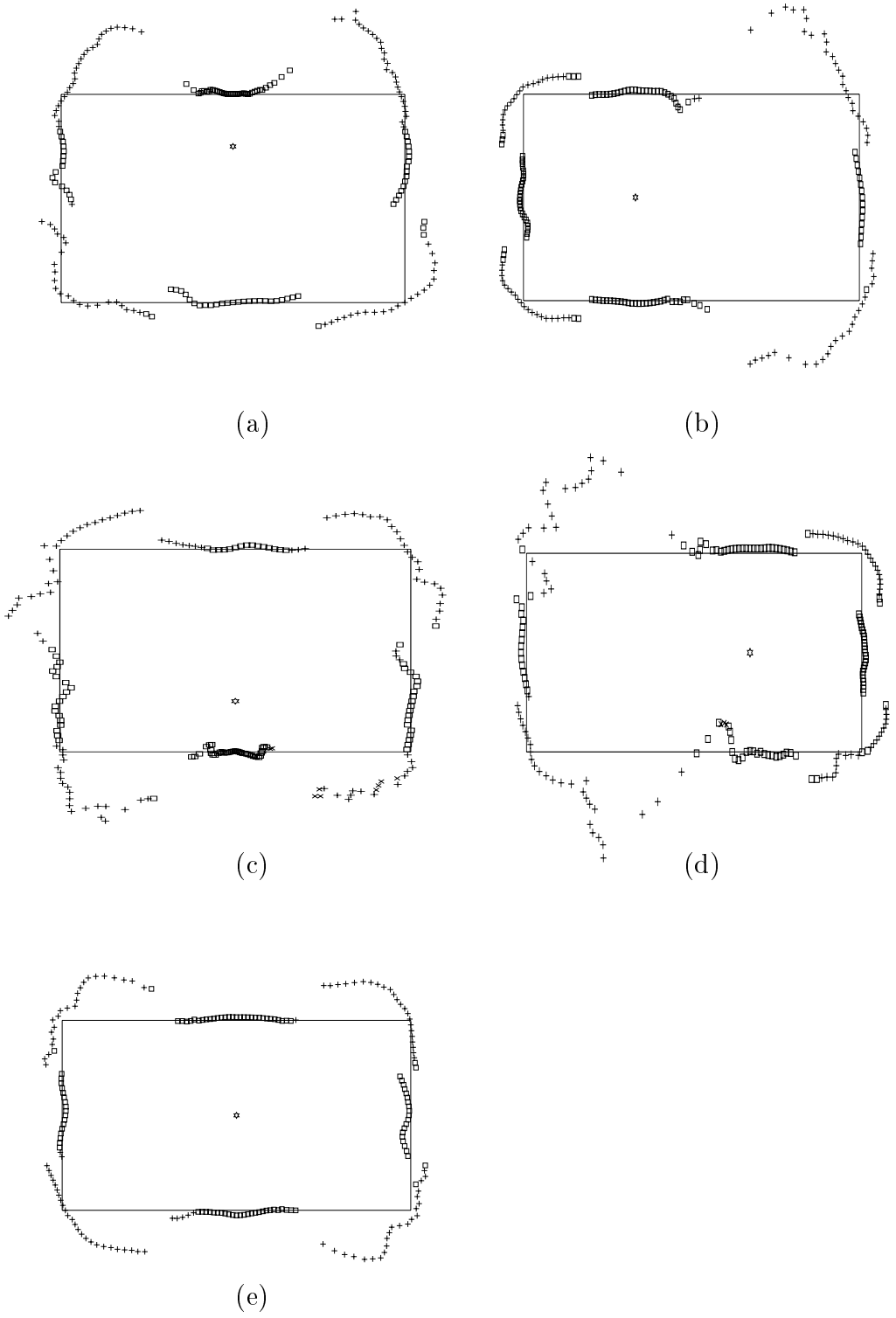


Figure 8.5: The local maps of room 1 extracted by employing newly-trained modular neural network classifier at the meet points (a) 1, (b) 2, (c) 3, (d) 4, and (e) 5.

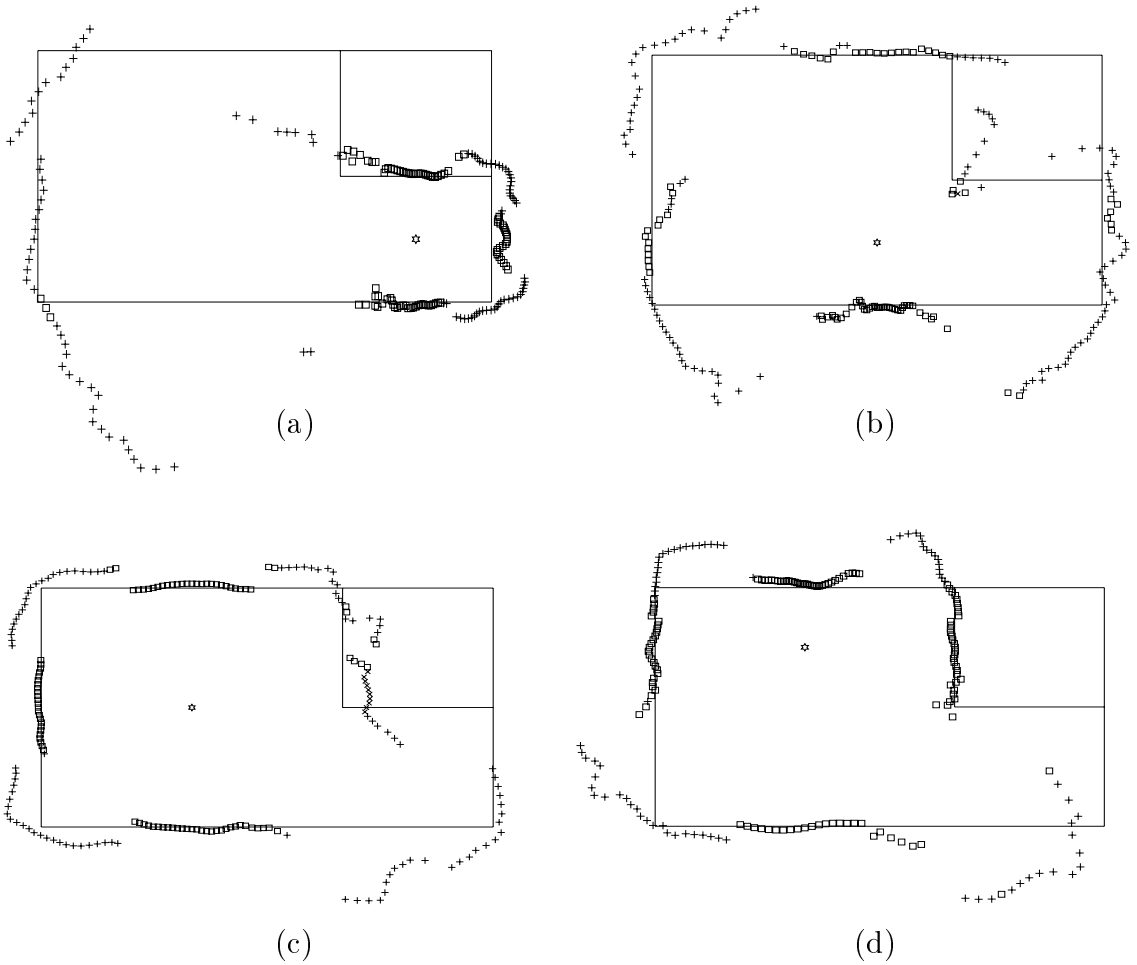


Figure 8.6: The local maps of room 2 extracted by employing newly-trained modular neural network classifier at the meet points (a) 1, (b) 2, (c) 3, and (d) 4.

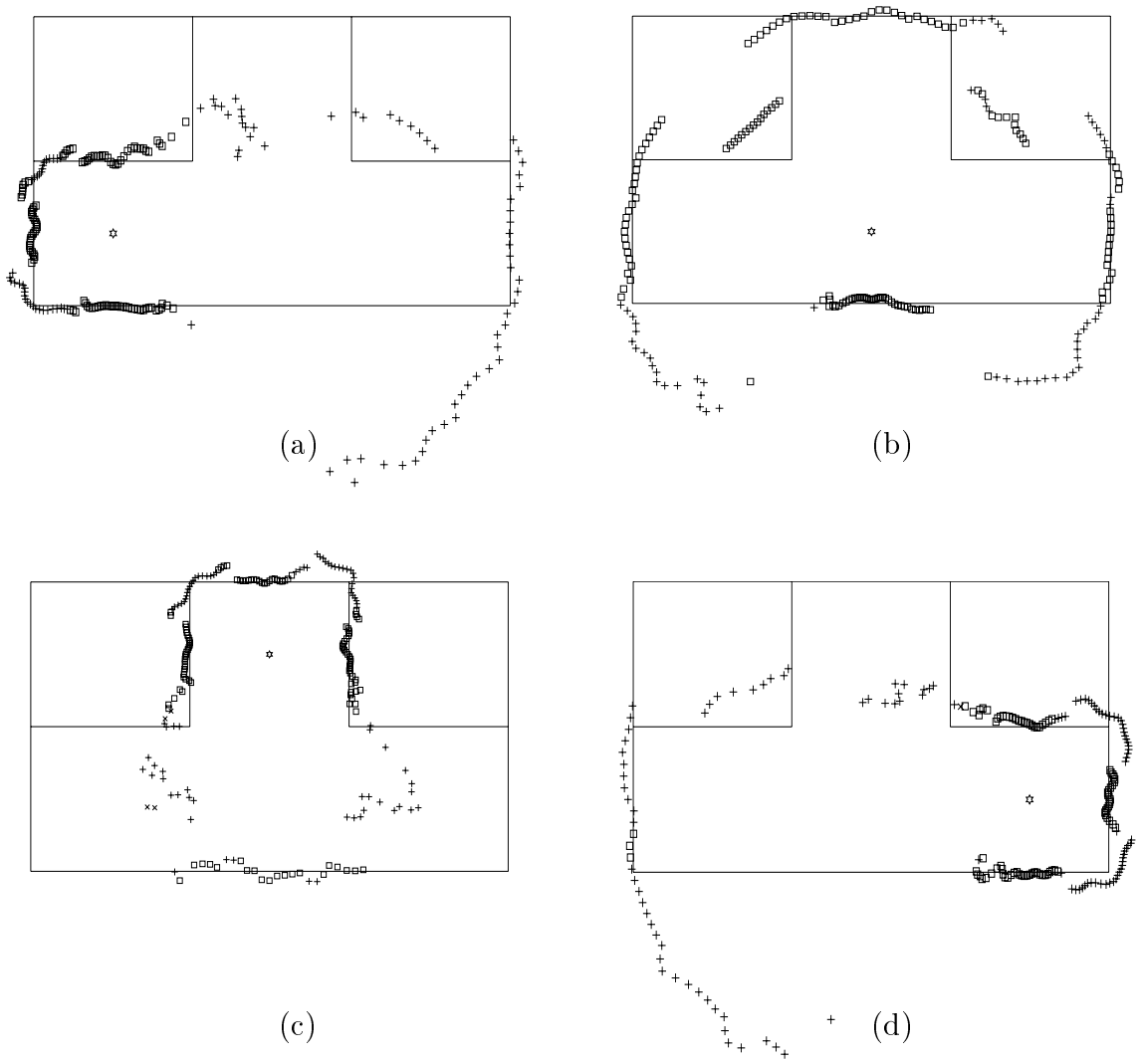


Figure 8.7: The local maps of room 3 extracted by employing newly-trained modular neural network classifier at the meet points (a) 1, (b) 2, (c) 3, and (d) 4.

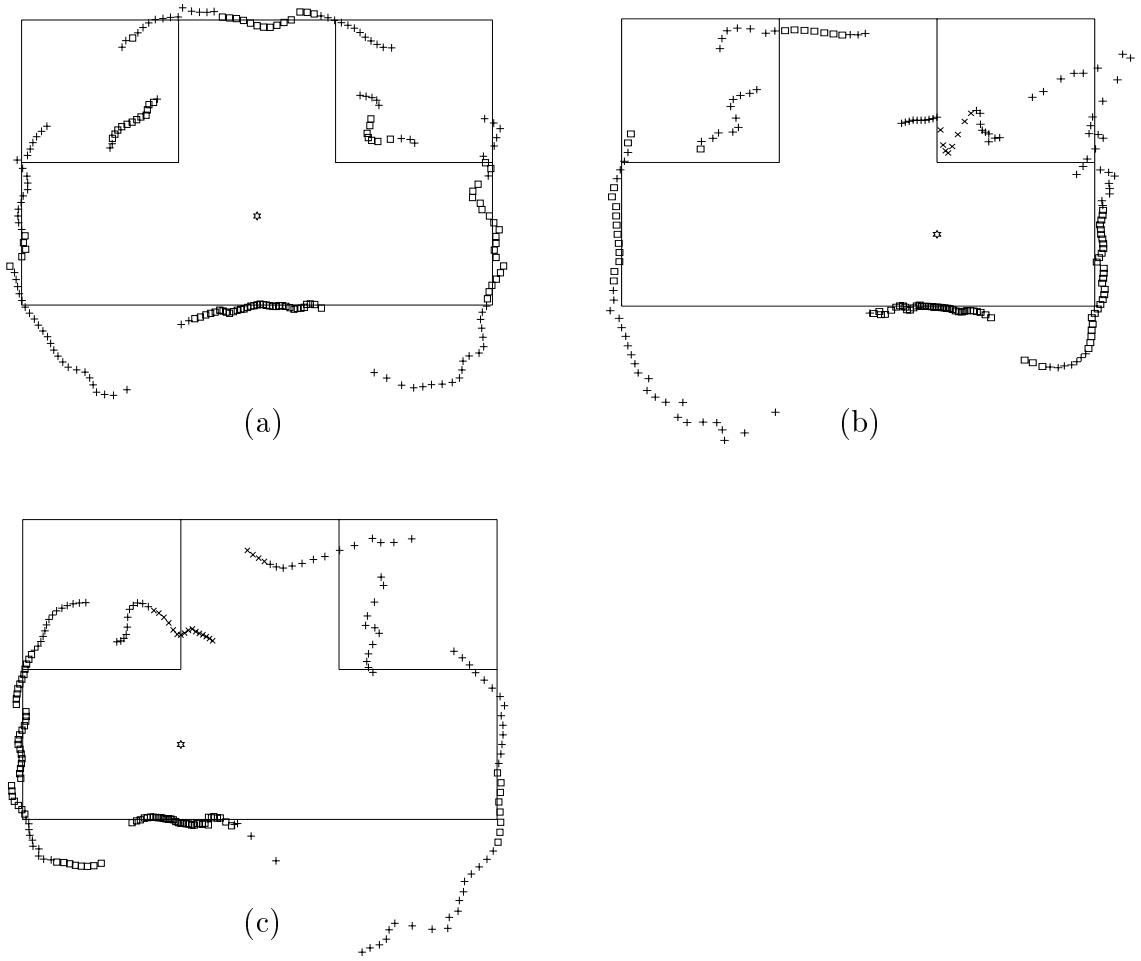


Figure 8.8: The local maps of room 3 extracted by employing newly-trained modular neural network classifier at the meet points (a) 5, (b) 6, and (c) 7.

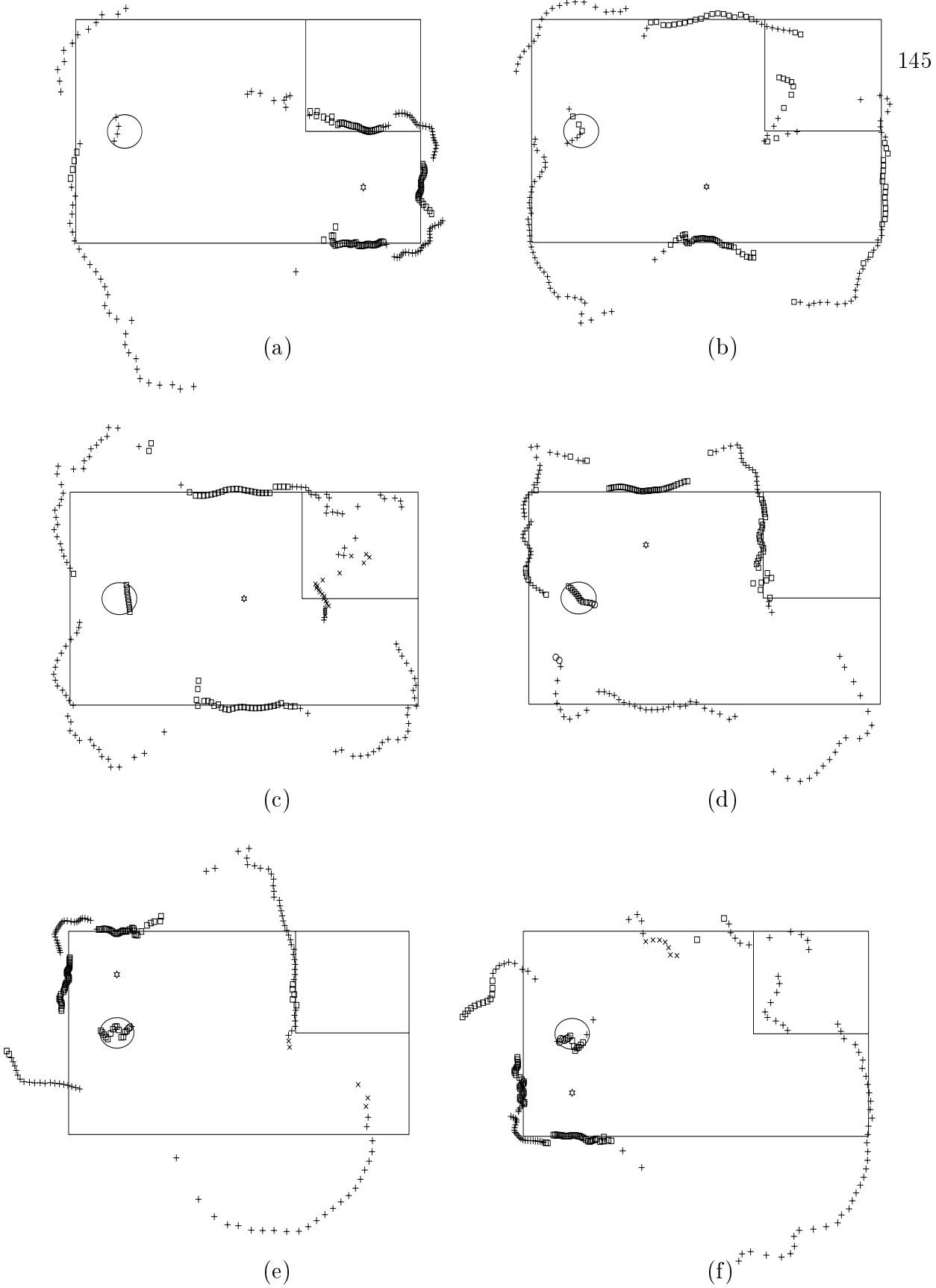


Figure 8.9: The local maps of room 4 extracted by employing newly-trained modular neural network classifier at the meet points (a) 1, (b) 2, (c) 3, (d) 4, (e) 5, and (f) 6.

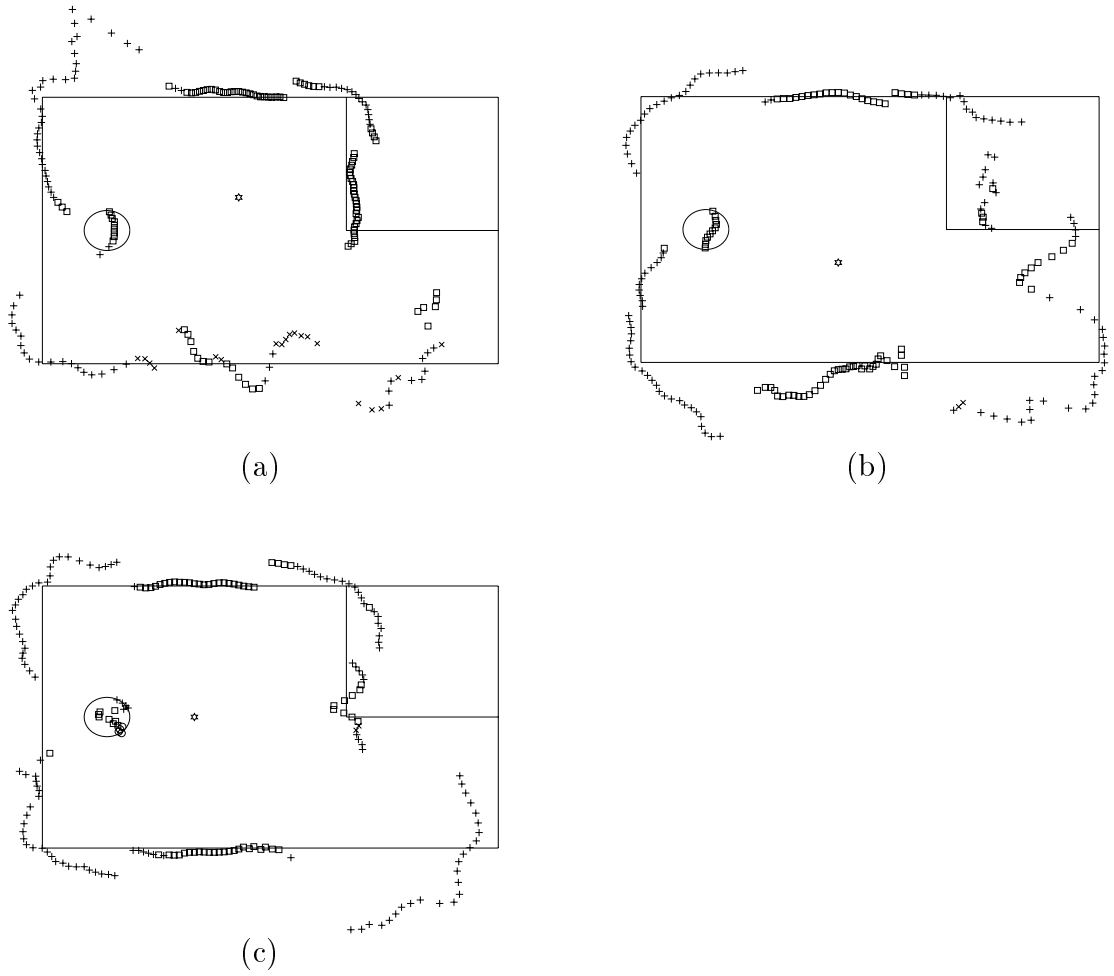


Figure 8.10: The local maps of room 4 extracted by employing newly-trained modular neural network classifier at the meet points (a) 7, (b) 8, and (c) 9.

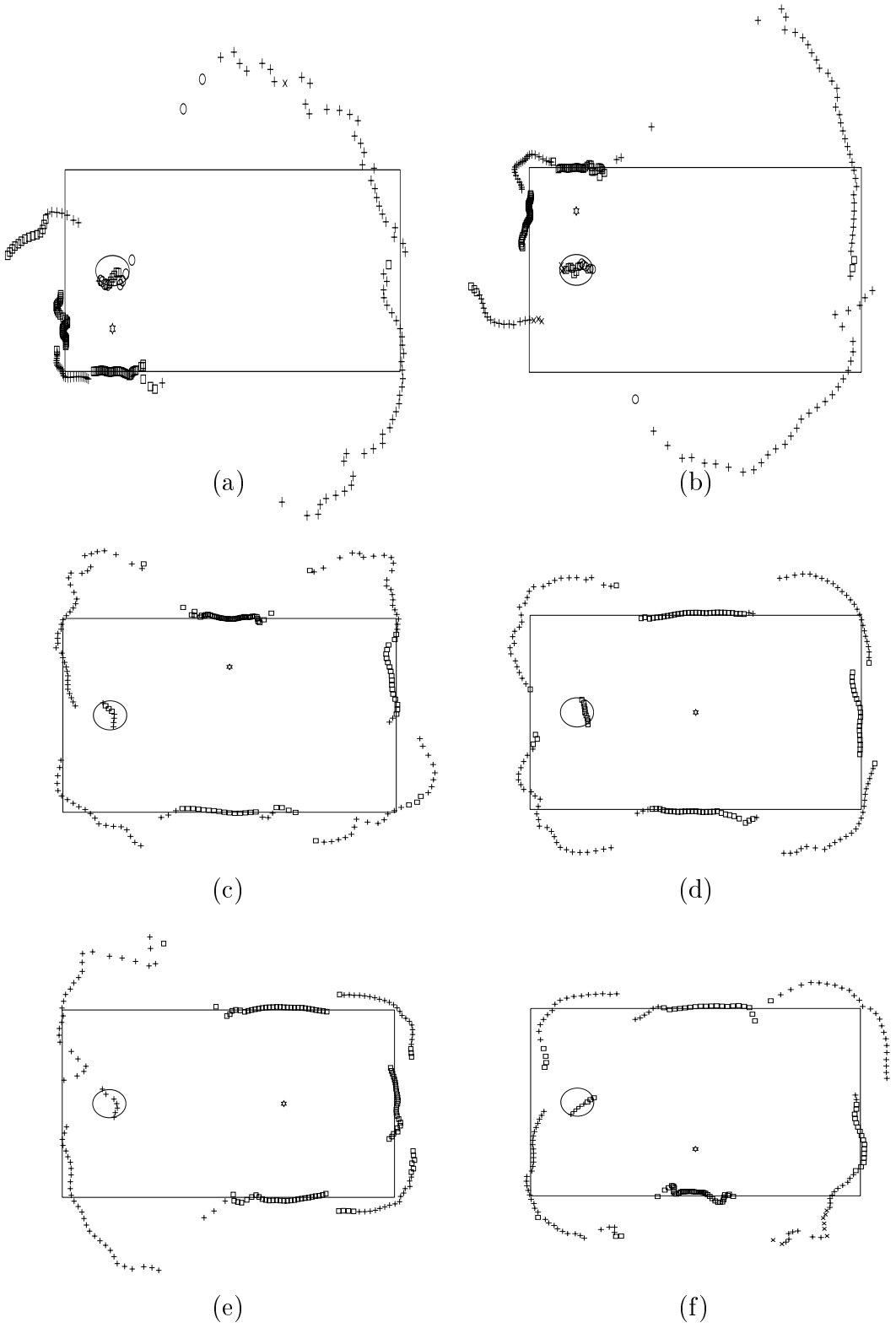


Figure 8.11: The local maps of room 5 extracted by employing newly-trained modular neural network classifier at the meet points (a) 1, (b) 2, (c) 3, (d) 4, (e) 5, and (f) 6.

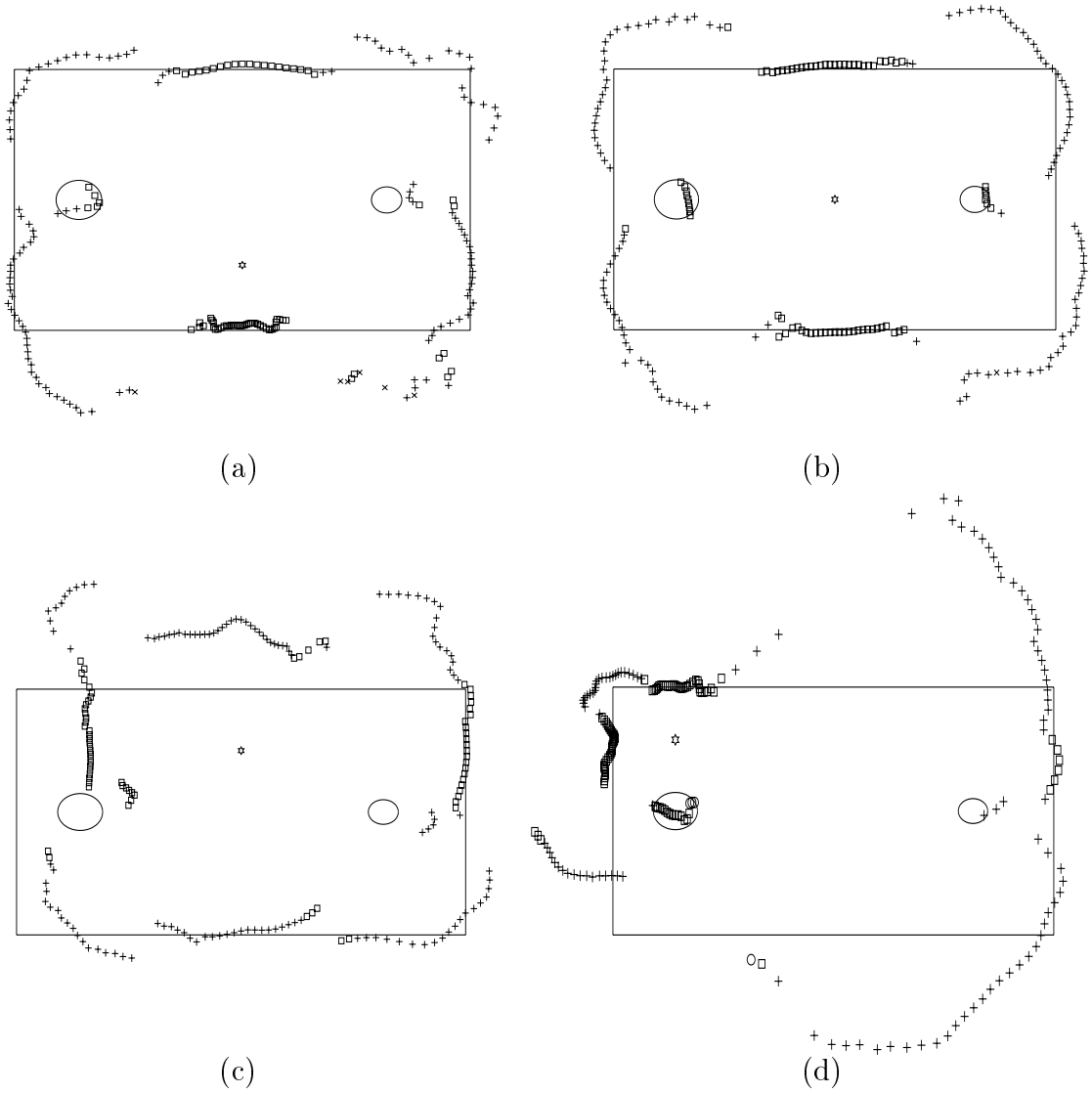
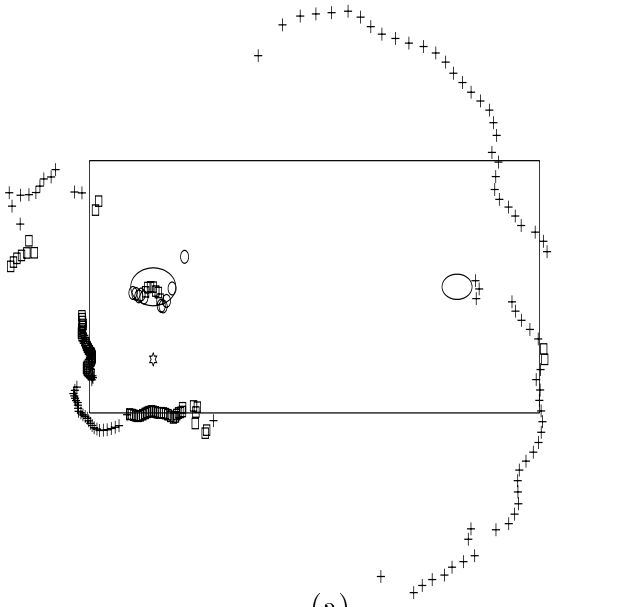
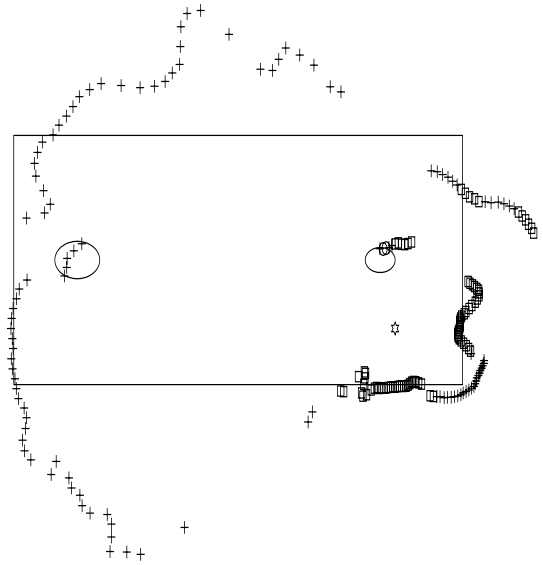


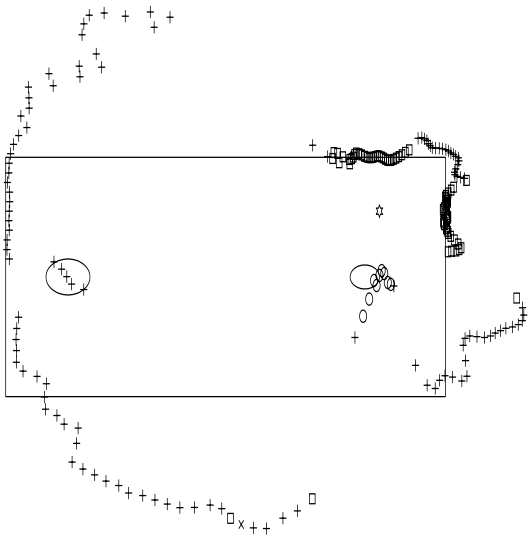
Figure 8.12: The local maps of room 6 extracted by employing newly-trained modular neural network classifier at the meet points (a) 1, (b) 2, (c) 3, and (d) 4.



(a)



(b)



(c)

Figure 8.13: The local maps of room 6 extracted by employing newly-trained modular neural network classifier at the meet points (a) 5, (b) 6, and (c) 7.

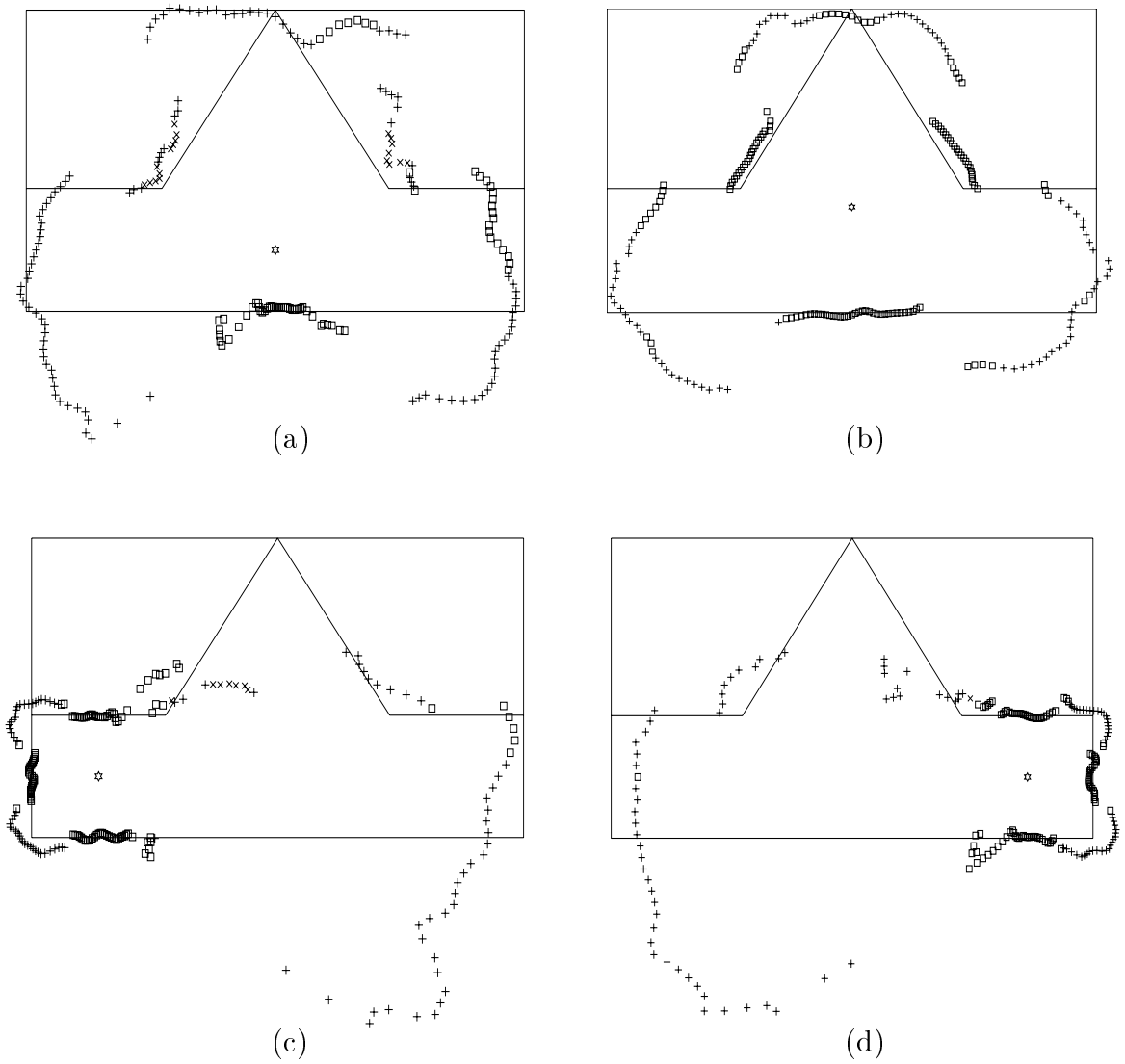


Figure 8.14: The local maps of room 7 extracted by employing newly-trained modular neural network classifier at the meet points (a) 1, (b) 2, (c) 3, and (d) 4.

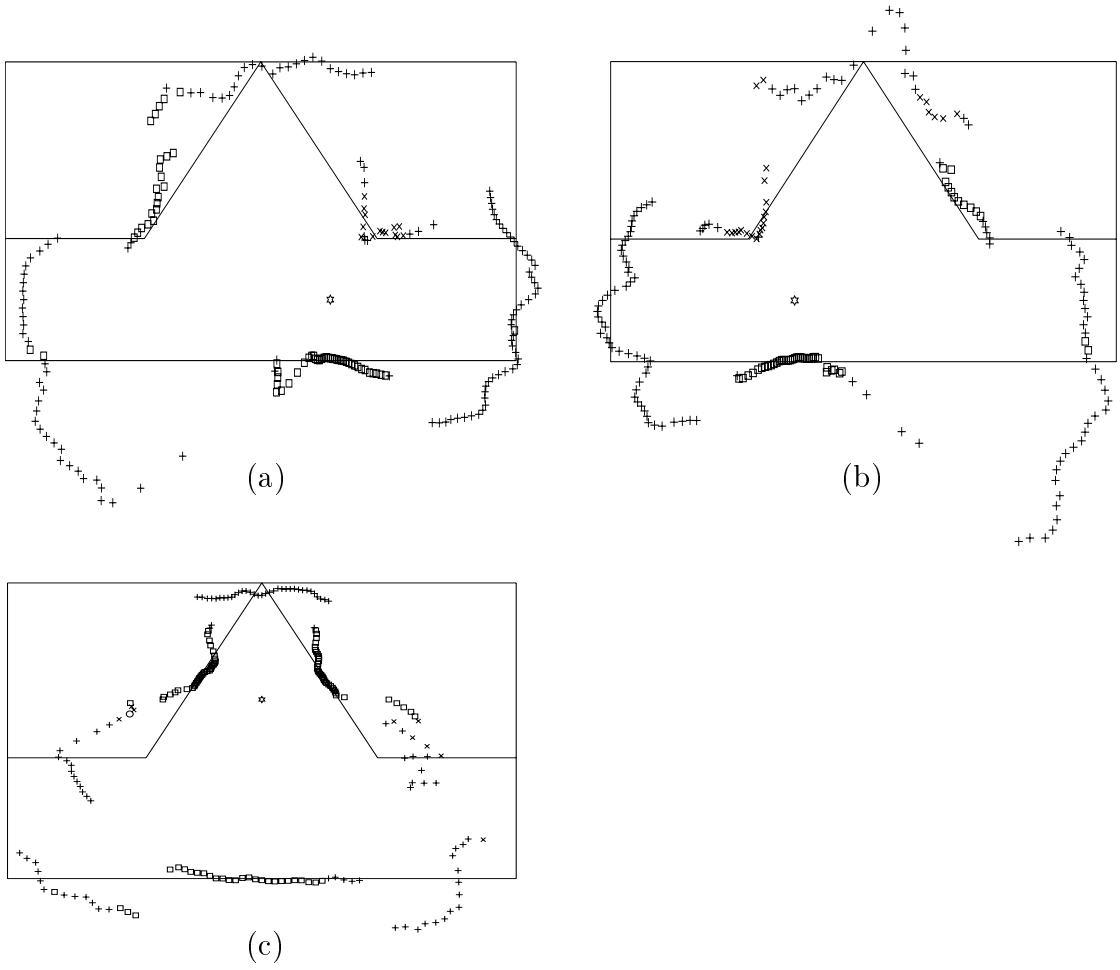


Figure 8.15: The local maps of room 7 extracted by employing newly-trained modular neural network classifier at the meet points (a) 5, (b) 6, and (c) 7.

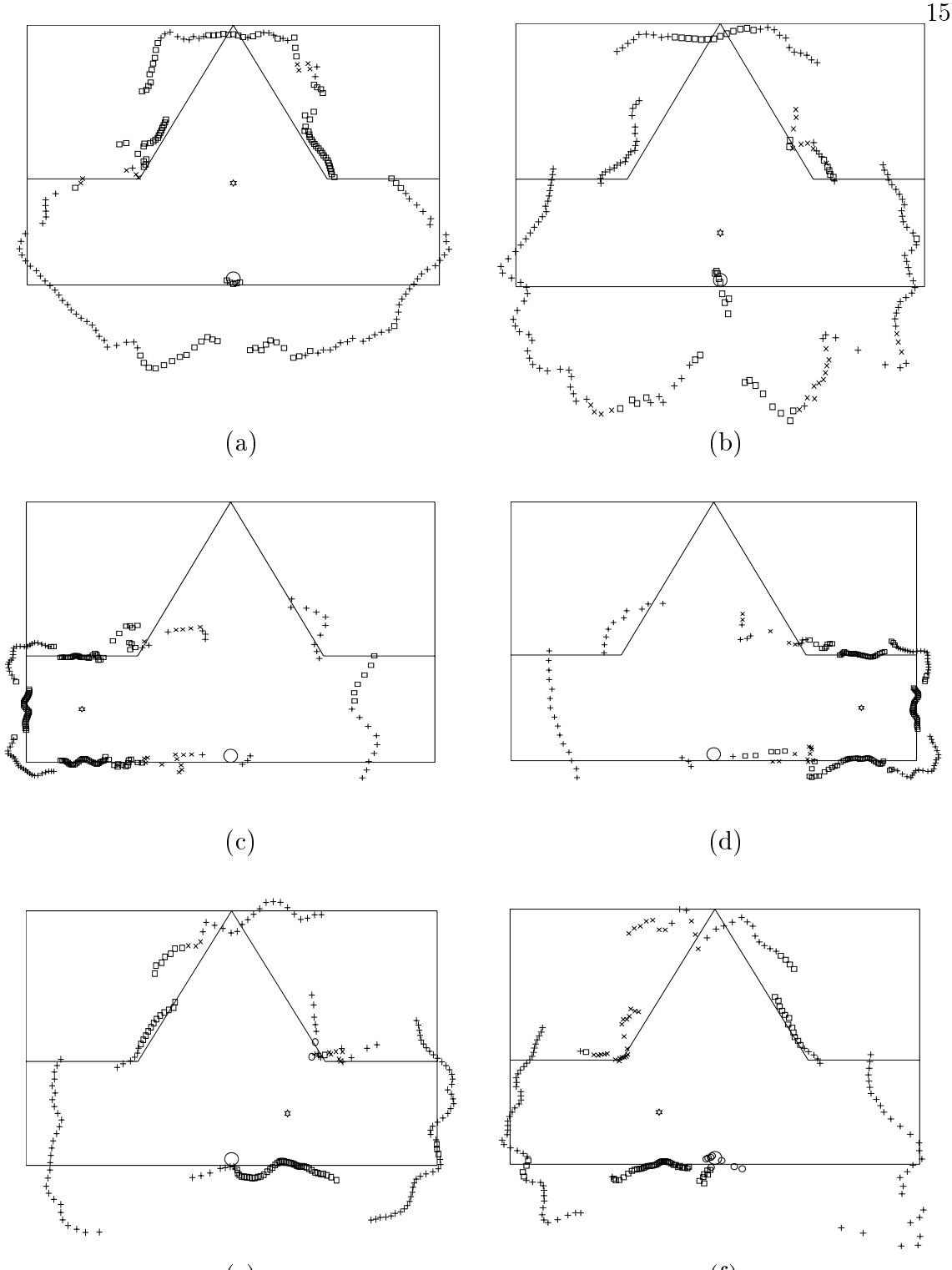


Figure 8.16: The local maps of room 8 extracted by employing newly-trained modular neural network classifier at the meet points (a) 1, (b) 2, (c) 3, (d) 4, (e) 5, and (f) 6.

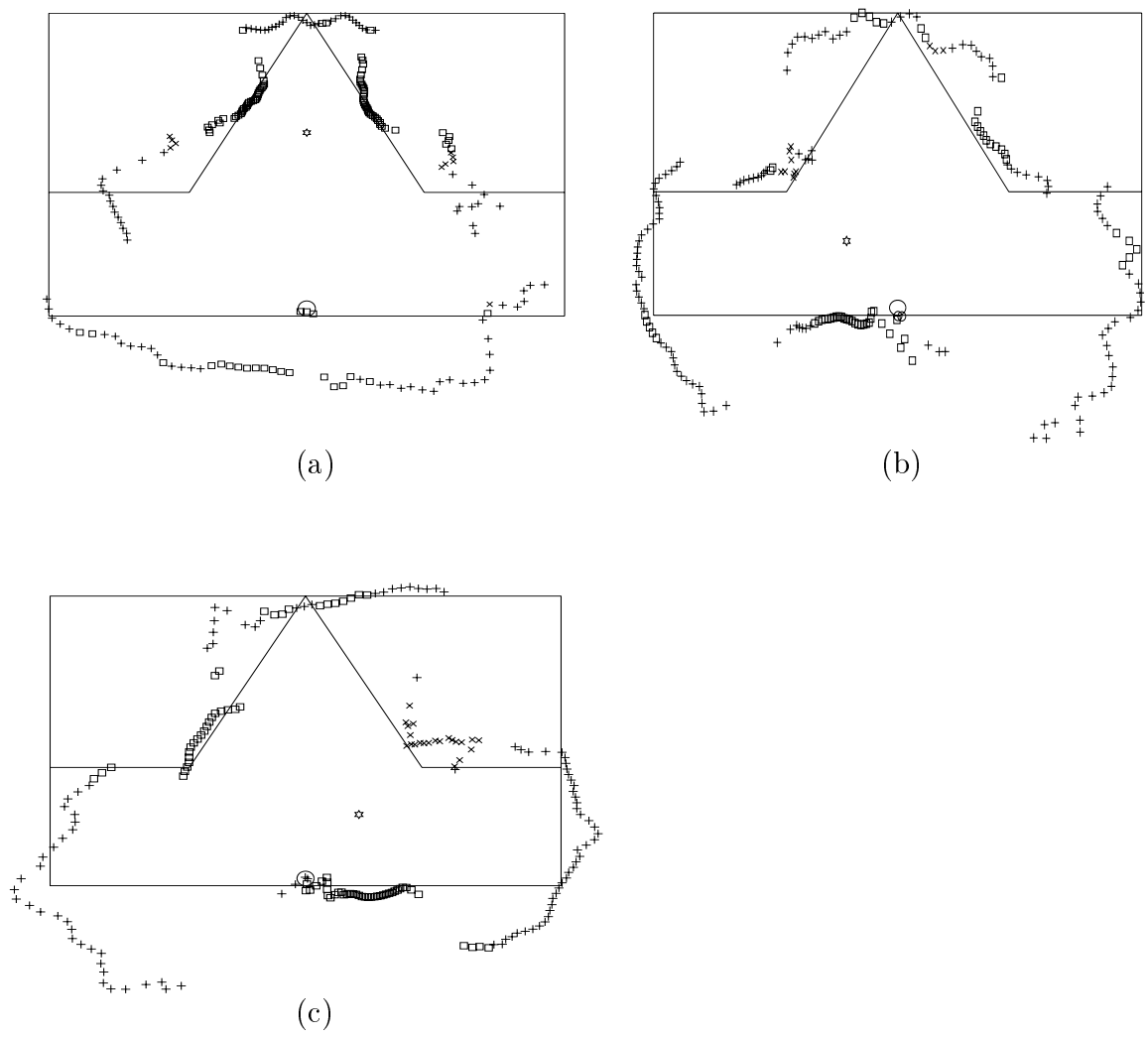


Figure 8.17: The local maps of room 8 extracted by employing newly-trained modular neural network classifier at the meet points (a) 7, (b) 8, and (c) 9.

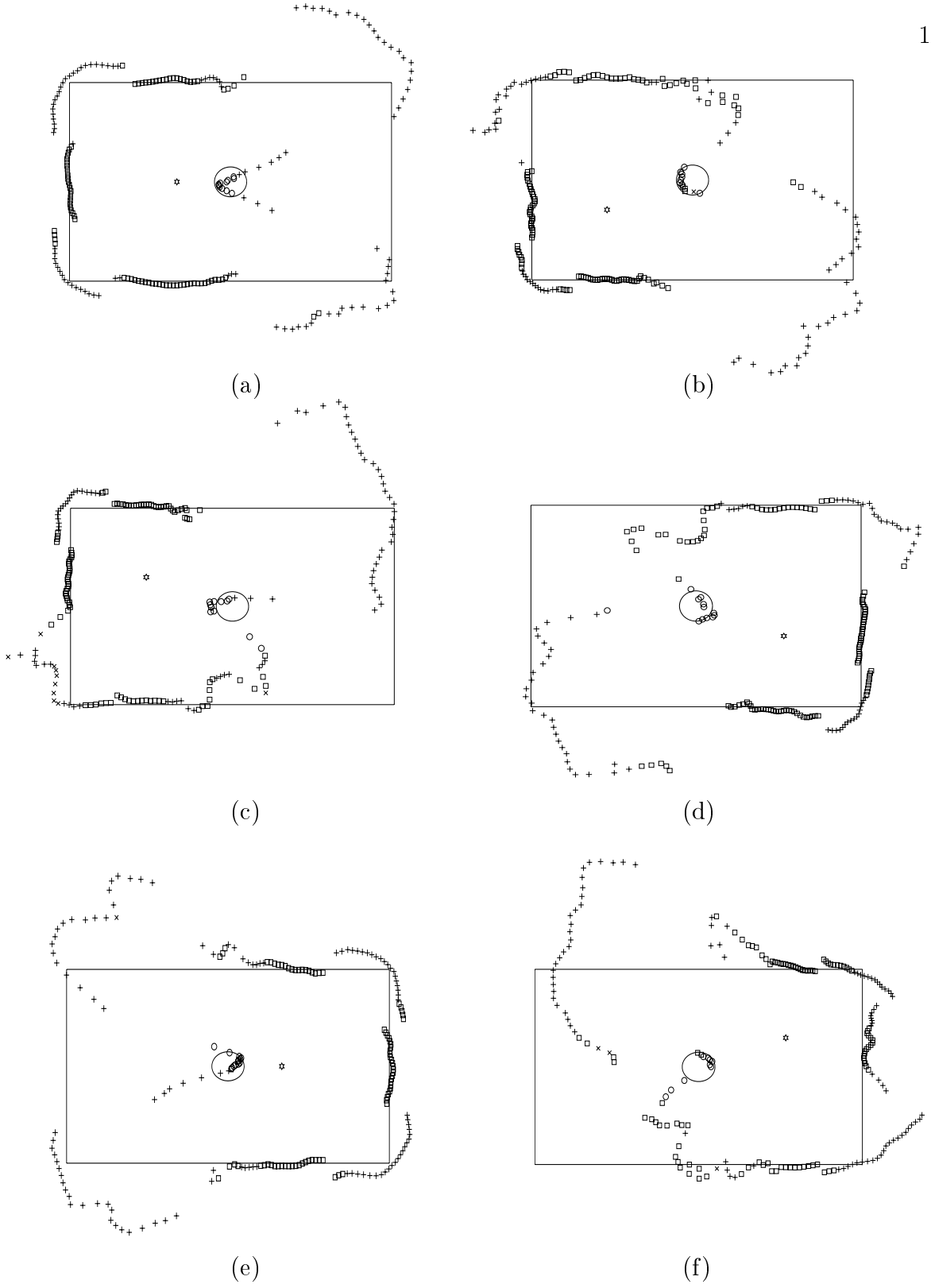
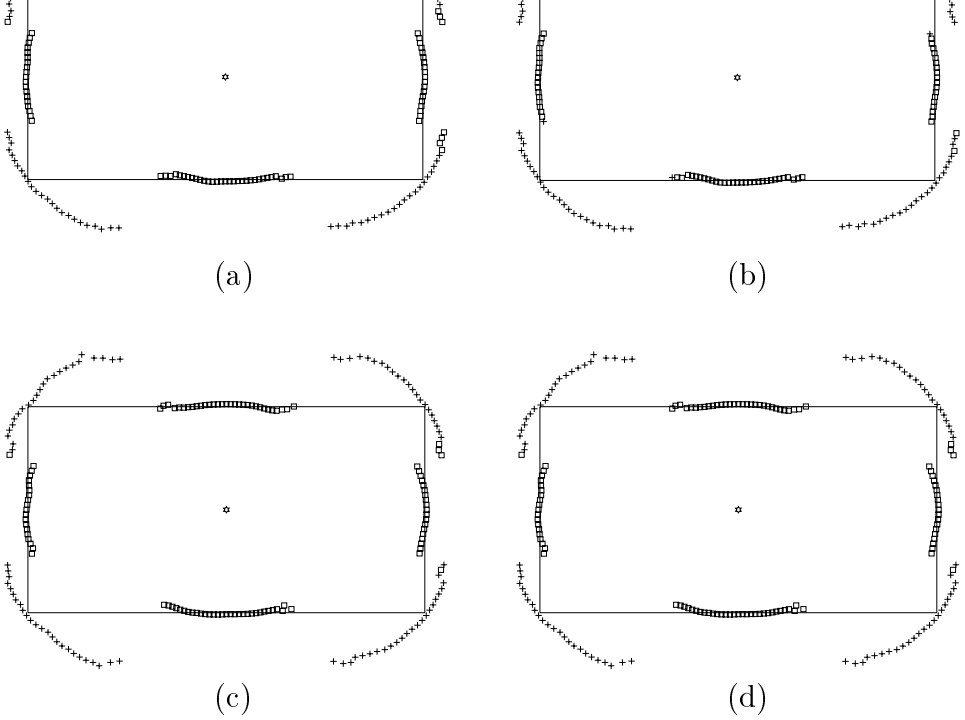


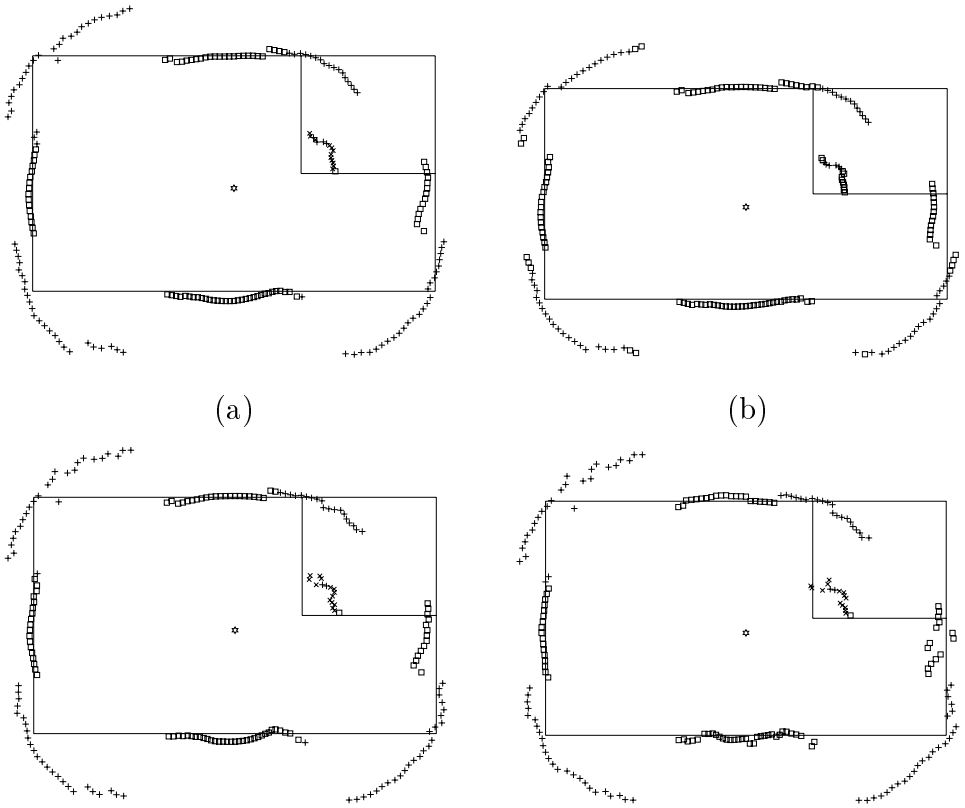
Figure 8.18: The local maps of room 9 extracted by employing newly-trained modular neural network classifier at the meet points (a) 1, (b) 2, (c) 3, (d) 4, (e) 5, and (f) 6.



(a) (b)

(c) (d)

Figure 8.19: The global maps of room 1 with respect to the centroid of its meet points extracted by employing (a) D-S, (b) SMV, (c) VRM, and (d) k-VRM.



(a) (b)

(c) (d)

Figure 8.20: The global maps of room 2 with respect to the centroid of its meet points extracted by employing (a) D-S, (b) SMV, (c) VRM, and (d) k-VRM.

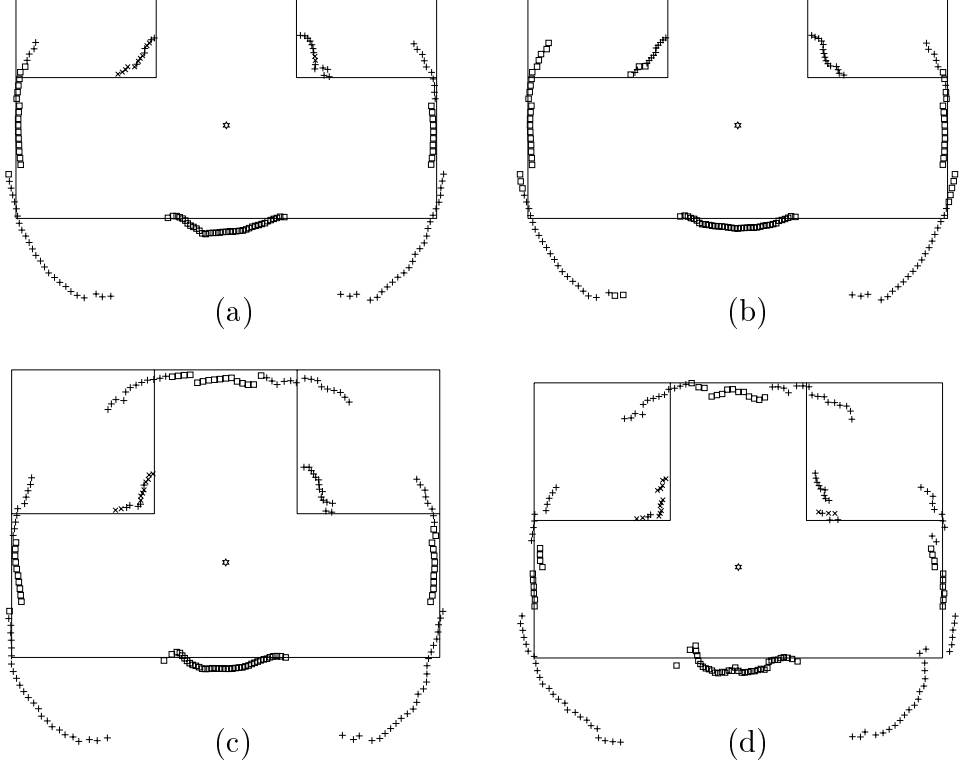


Figure 8.21: The global maps of room 3 with respect to the centroid of its meet points extracted by employing (a) D-S, (b) SMV, (c) VRM, and (d) k-VRM.

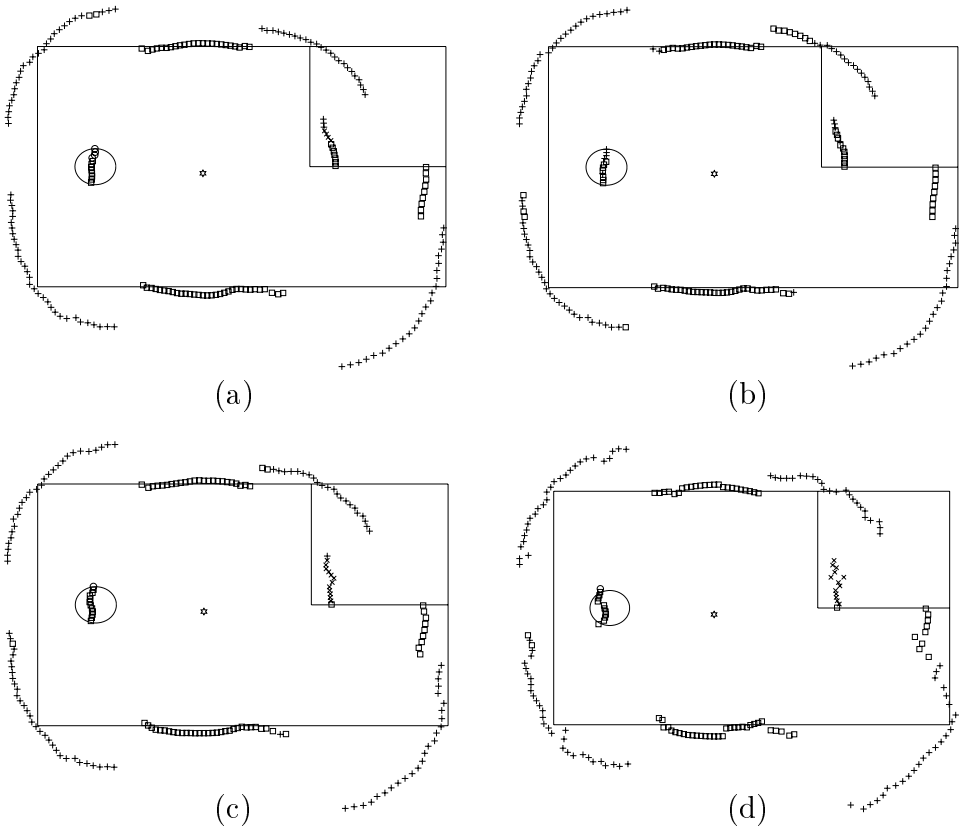


Figure 8.22: The global maps of room 4 with respect to the centroid of its meet points extracted by employing (a) D-S, (b) SMV, (c) VRM, and (d) k-VRM.

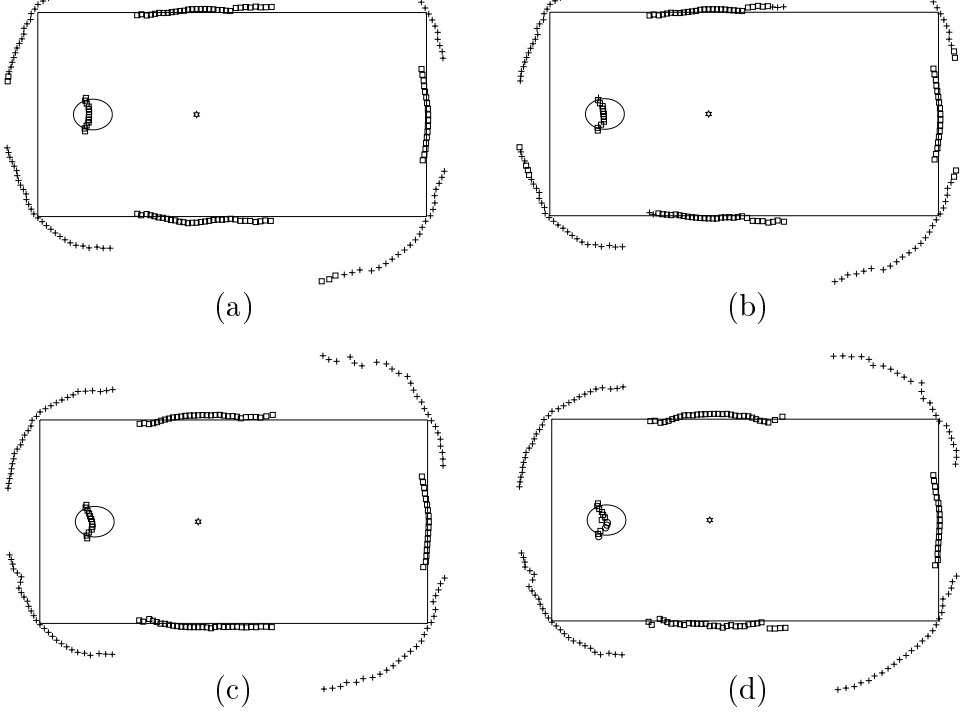


Figure 8.23: The global maps of room 5 with respect to the centroid of its meet points extracted by employing (a) D-S, (b) SMV, (c) VRM, and (d) k-VRM.

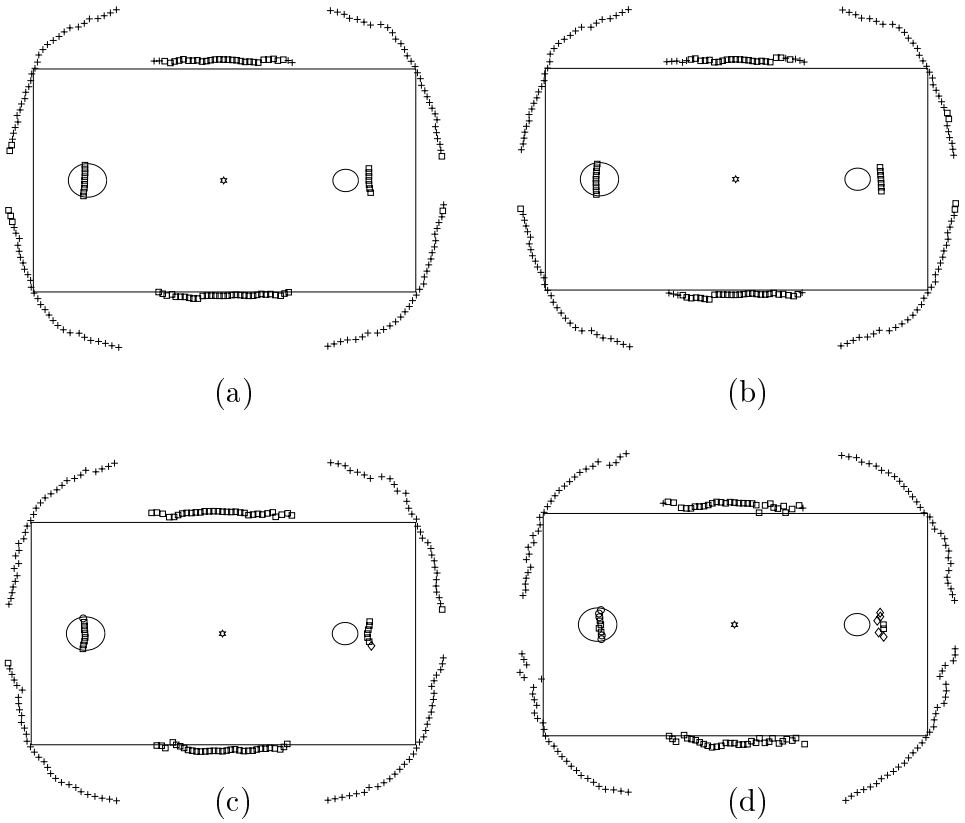


Figure 8.24: The global maps of room 6 with respect to the centroid of its meet points extracted by employing (a) D-S, (b) SMV, (c) VRM, and (d) k-VRM.

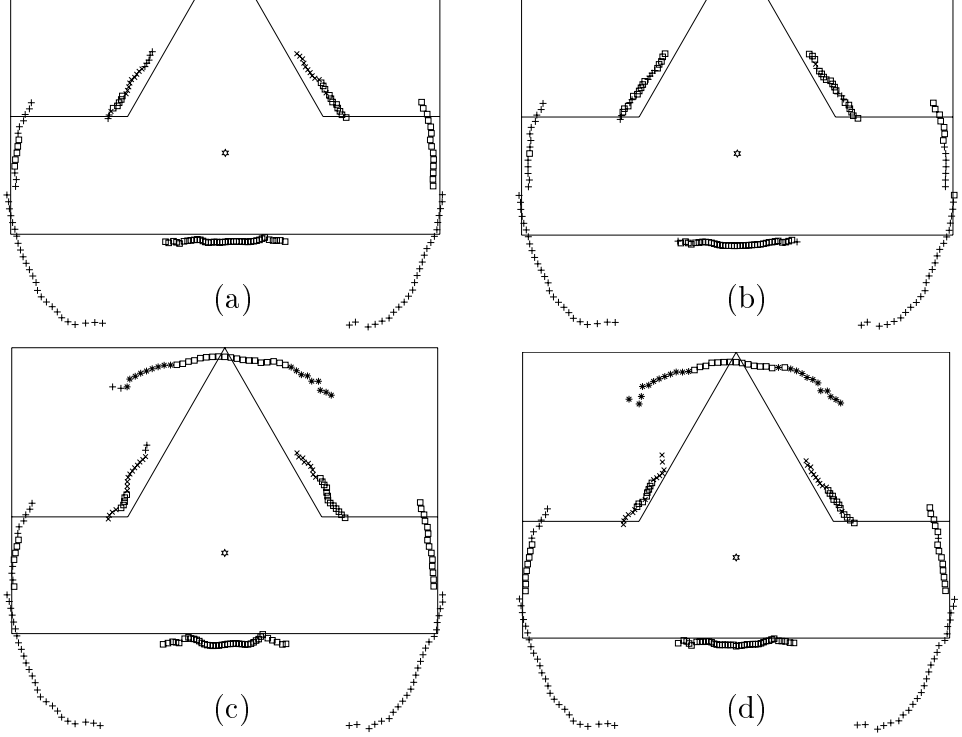


Figure 8.25: The global maps of room 7 with respect to the centroid of its meet points extracted by employing (a) D-S, (b) SMV, (c) VRM, and (d) k-VRM.

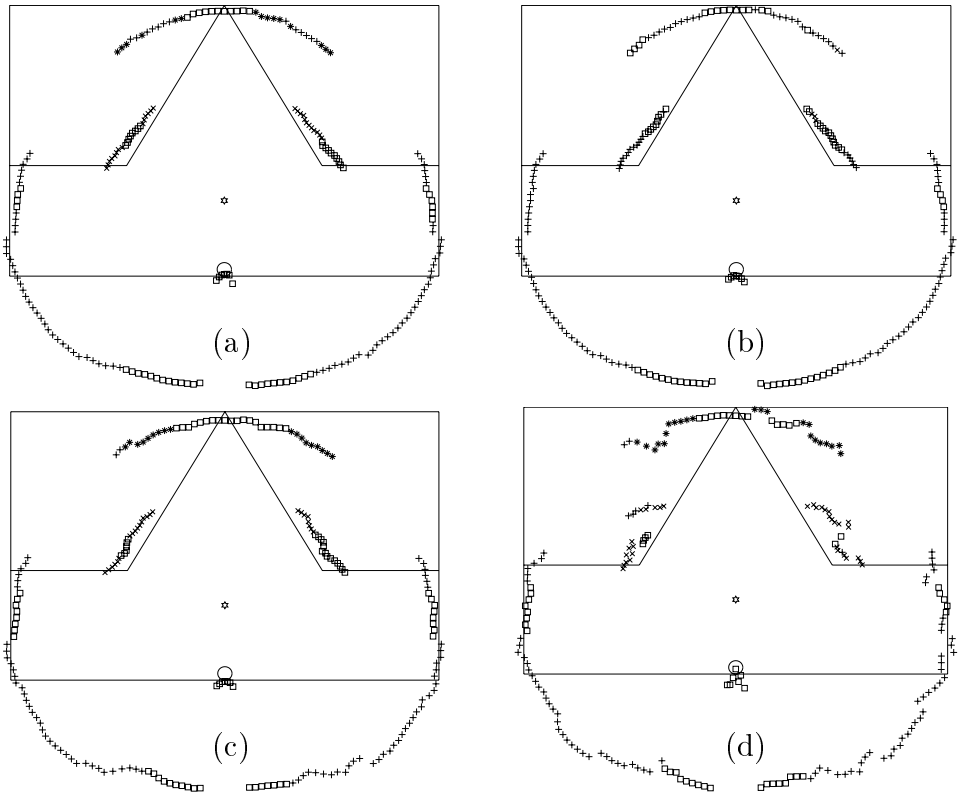


Figure 8.26: The global maps of room 8 with respect to the centroid of its meet points extracted by employing (a) D-S, (b) SMV, (c) VRM, and (d) k-VRM.

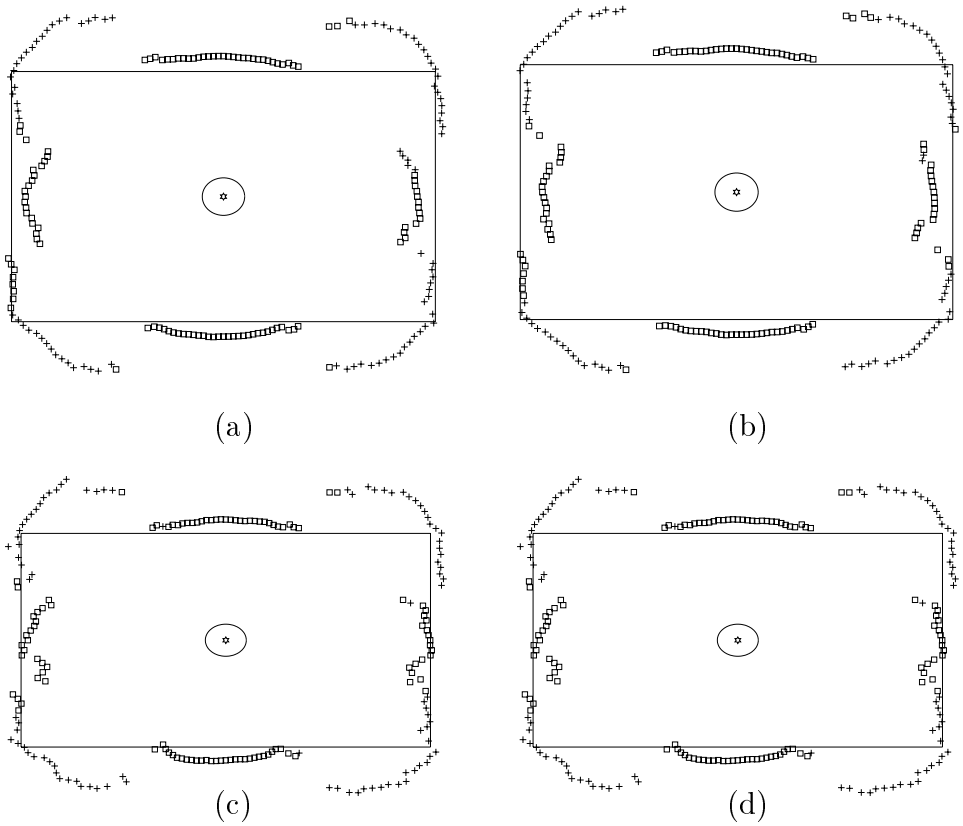


Figure 8.27: The global maps of room 9 with respect to the centroid of its meet points extracted by employing (a) D-S, (b) SMV, (c) VRM, and (d) k-VRM.

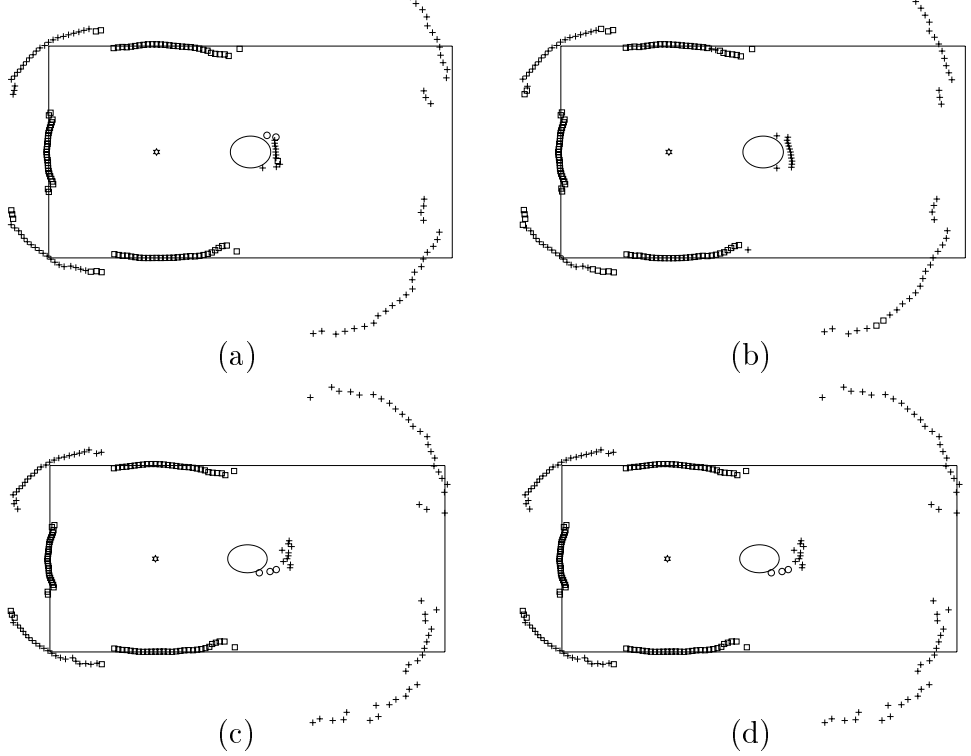


Figure 8.28: The global maps of room 9 with respect to the left centroid of its meet points extracted by employing (a) D-S, (b) SMV, (c) VRM, and (d) k-VRM.

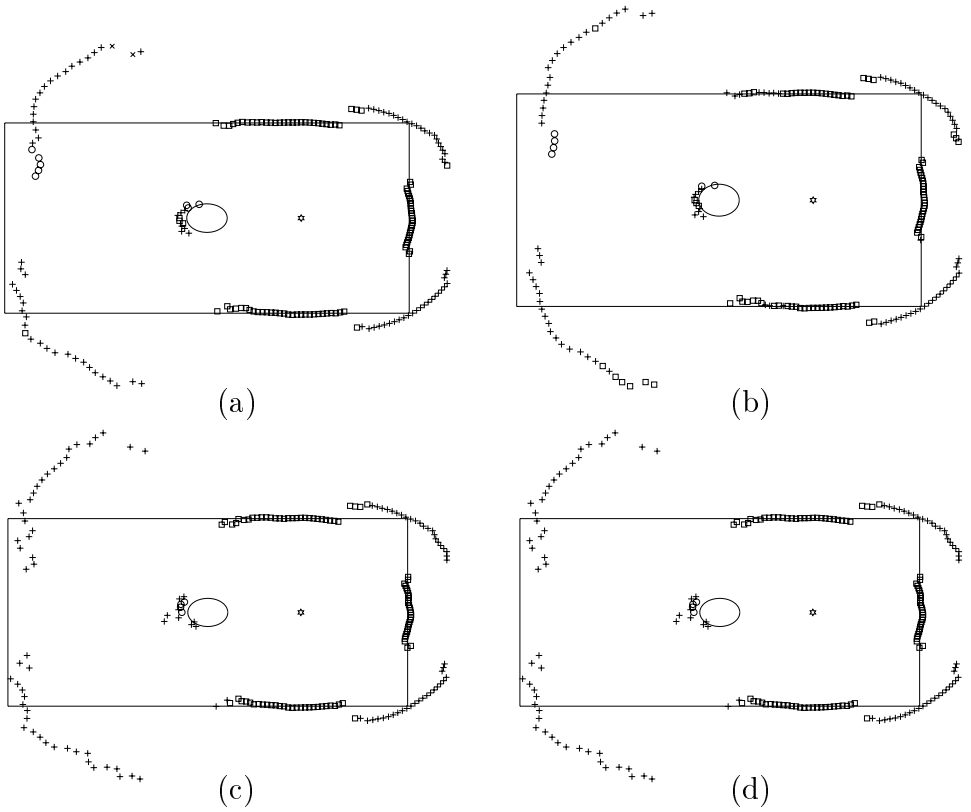


Figure 8.29: The global maps of room 9 with respect to the right centroid of its meet points extracted by employing (a) D-S, (b) SMV, (c) VRM, and (d) k-VRM.

Chapter 9

INCLUSION OF PHYSICALLY DIFFERENT SENSORS

In this chapter, inclusion of physically different sensors besides sonars in the classification of targets existing in a mobile robot environment and extraction a feature-based map of the environment is described.

This chapter is organized as follows: Nomad 200^{TM} mobile robot which is used in the experiments is introduced in Section 9.1. In Section 9.2, physically different sensors besides sonars are employed in the target classification problem and the maps of mobile robot environments are extracted experimentally with these physically different sensors.

9.1 Nomad 200^{TM}

The Nomad 200^{TM} mobile robot represented in Figure 9.1 is employed in the experiments. It consists of tactile, infrared, sonar, and structured-light sensing systems.

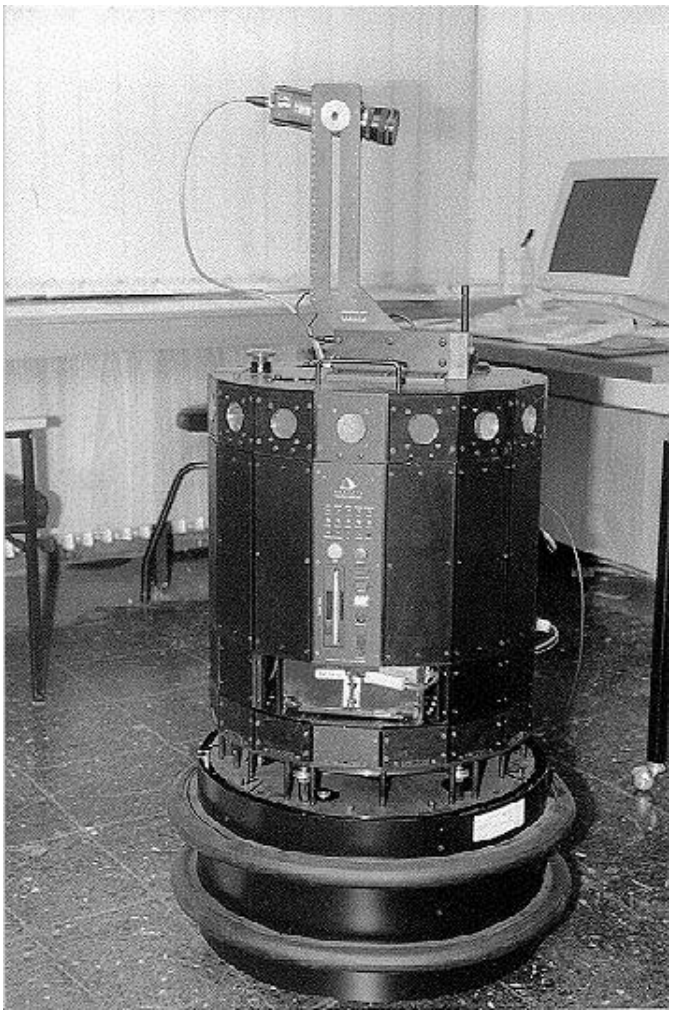


Figure 9.1: The Nomad 200^{TM} mobile robot.

Its height and diameter are 120.0 and 45.7 cm, respectively. The base translation, base rotation, and turret rotation are controlled by three separate motors. The robot can translate only in the forward and backward directions. In order to translate sideways, it needs to rotate first. The maximum translational and rotational speeds of the robot are 60 cm/s and 60°/s, respectively.

The Sensus 100 Tactile System consists of 20 independent pressure sensitive tactile sensors to detect contact with an obstacle in the environment. This sensing modality is not used in this study.

The Sensus 200 Sonar Ranging System contains 16 Polaroid 6500 series ultrasonic transducers [13] to obtain range information via measuring the TOF. These sensors can provide range values between 15 cm and 10.7 m with $\pm 1\%$ accuracy. Their beamwidths ($2\theta_\circ$) are 25° and resonance frequencies ($f_\circ$) are 49.4 kHz.

The Sensus 300 Infrared Proximity System consists of 16 infrared sensors to provide range information up to 60 cm by measuring the intensity of the light reflected back to the detector by an object. Each infrared sensor has 2 LED emitters and a photodiode detector.

The Sensus 500 contains a laser diode and a CCD camera which is mostly sensitive to the laser frequency (corresponding to 685 nm wavelength). The operating range of the system is between 30.5 cm and 3.05 m. In order to obtain a plane of light, laser beam is passed through a cylindrical lens. The intersection of the plane of light with an object within the operating range of the system is detected by the camera with the help of an interference filter. In this system, the range value is calculated by using the triangulation technique which is characterized by reduced accuracy with increasing range.

9.2 Experimental Studies

13 different data sets collected at Bilkent University Robotics Research Laboratory by scanning three different experimental test areas with the Nomad 200^{TM} mobile robot are provided to the author. The mobile robot Nomad 200^{TM} navigates in these rooms by using the wall-following algorithm. During this operation, infrared and sonar data are collected by activating the three sonar and infrared sensors at the direction perpendicular to the moving direction of the robot represented in Figure 9.2. The structured-light system available on the Nomad 200^{TM} is activated to collect laser data at the same time.

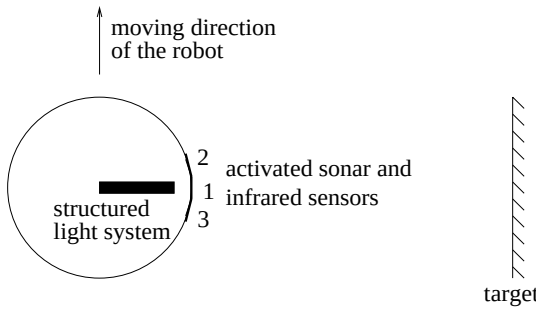


Figure 9.2: The positions of the structured-light system and the three activated sonar and infrared sensors on the Nomad 200^{TM} mobile robot with respect to the moving direction of the robot.

Initially, an algorithm using only sonar data for the classification of targets existing in the Nomad 200^{TM}'s environment and extraction a feature-based map of that environment is developed. This algorithm is referred as Algorithm I throughout this chapter. Using this algorithm, plane, corner, edge, and cylinder can be differentiated and the feature-based map of the environment is extracted by specifying the target in front of the robot. In Algorithm I, the difference in range data taken from the sonar sensor at the center and the one to its right, derivative of this differential signal, and the derivative of the range data taken from the sonar sensor at the center are used. These signals are referred as $x(n)$, $y(n)$, and $z(n)$ respectively. The derivative of these signals are calculated by using backward differentiation such that $y(n) = \frac{x(n)-x(n-1)}{1}$. Based on close inspection of the data, an algorithm is developed which can be summarized in the form of rules as follows:

Algorithm I:

If $|x(n)| \leq 9$ cm

if $|y(n)| > 6$ cm then $t(n)$= **corner**

if $|y(n)| \leq 6$ cm then $t(n)$= **plane**.

If $|x(n)| > 9$ cm then $t(n)$= **corner**

if $z(n) = 0$ cm then $t(n)$= **plane**

if $z(n) = 0$ cm and $x(n) < 0$ then $t(n)$= **cylinder**.

If $t(n-1)$= plane and $t(n)$= corner and $z(n) > 20$ cm then $t(n)$= **edge**.

To improve the performance of target classification, infrared sensors are also employed. Another algorithm referred as Algorithm II throughout this chapter is developed which fuses the sonar and infrared sensor readings. In this algorithm, the difference in the intensity data taken from the infrared sensor at the center and the one to its right $x_r(n)$, and difference in the intensity data taken from the infrared sensor at the center and the one to its left $x_l(n)$ are included besides the sonar signals used in Algorithm I. Algorithm II can also be summarized in the form of rules as follows:

Algorithm II:

If $|x(n)| \leq 9$ cm

if $|y(n)| > 6$ cm then $t(n)$= **corner**

if $|y(n)| \leq 6$ cm then $t(n)$= **plane**.

If $|x(n)| > 9$ cm

if $|x_l(n)| > 0$ then $t(n)$= **corner**

else if $|x_r(n)| < 5.5$ and $x(n) < 0$ then $t(n)$= **cylinder**

else if $z(n) = 0$ cm then $t(n)$= **plane**

else if $y(n) > 6$ cm then $t(n)$= **edge**

else $t(n)$= **unknown**.

Average percentages of correct classification achieved by employing Algorithm I and Algorithm II over the 13 data sets are given below. Note that Algorithm II cannot be applied to the first three data sets since infrared data are not available for these data sets. The classification performance of Algorithm I is increased by up to 5% using Algorithm II. Just for illustrative purposes, the sonar and infrared signals employed in Algorithm I and II for data set 12 are given in Figures 9.3−9.4.

Data Set	Algorithm I	Algorithm II
1	95 %	–
2	90 %	–
3	91 %	–
4	94 %	96 %
5	92 %	96 %
6	95 %	97 %
7	95 %	97 %
8	92 %	96 %
9	88 %	93 %
10	91 %	96 %
11	90 %	95 %
12	91 %	95 %
13	87 %	87 %

Finally, feature-based maps of the environments are extracted from these 13 data sets by specifying the target type identified by Algorithms I and II along the line-of-sight of the sonar sensor at the center. The best maps obtained by employing Algorithms I and II for three experimental test areas and the robot's path which is followed while extracting these maps are plotted in Figures 9.5–9.7. Readings collected by the structured-light system are very accurate and are taken as an absolute reference to be used for comparison. For this reason, they are not used in Algorithms I and II. As an example, robot's positions and the laser readings at these positions for data set 12 are illustrated in Figure 9.8. Referring to Figures 9.6 and 9.8, it can be easily seen that the range measurement of the sonar sensor at the center and the range determined by the structured-light system are comparable.

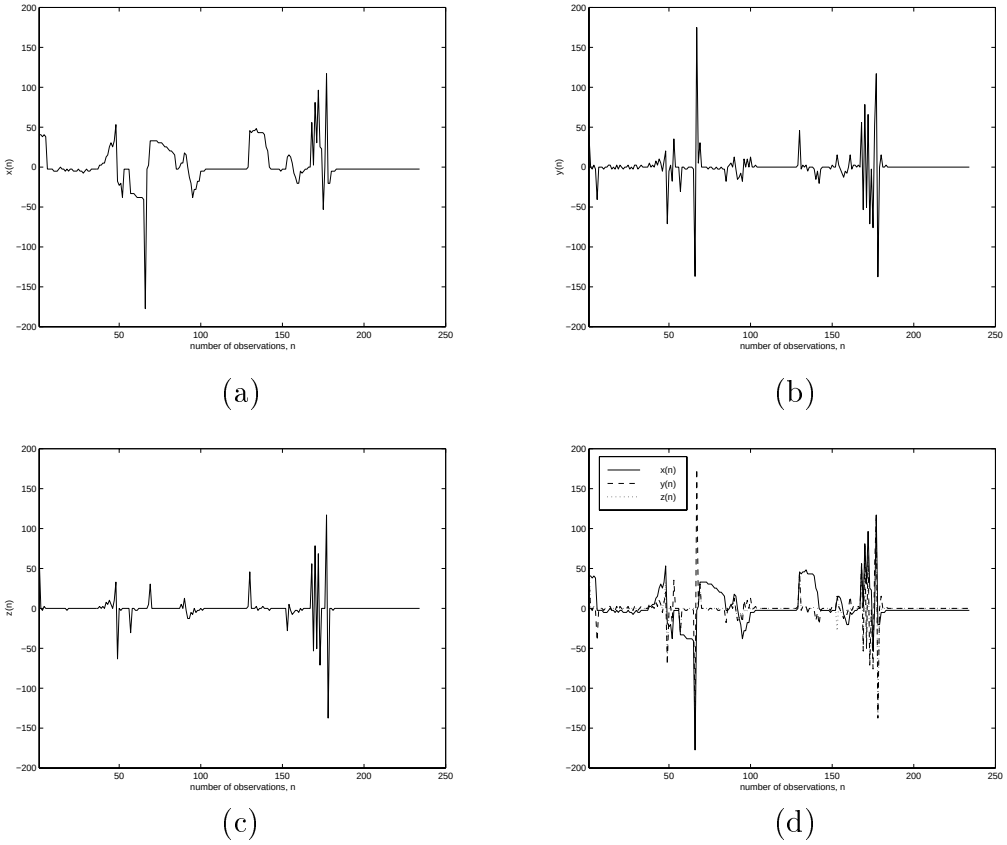


Figure 9.3: Sonar signals (a) $x(n)$ (b) $y(n)$ (c) $z(n)$ and (d) $x(n)$, $y(n)$, and $z(n)$ which are collected by Nomad 200^{TM} (data set 12).

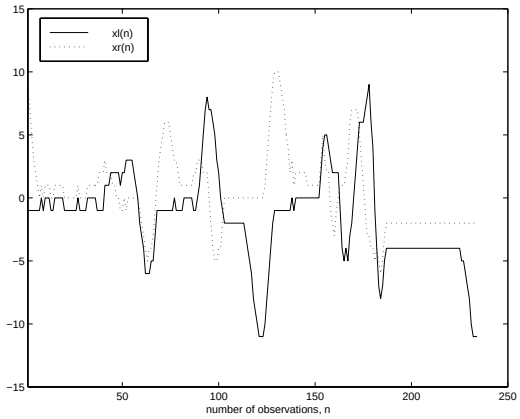
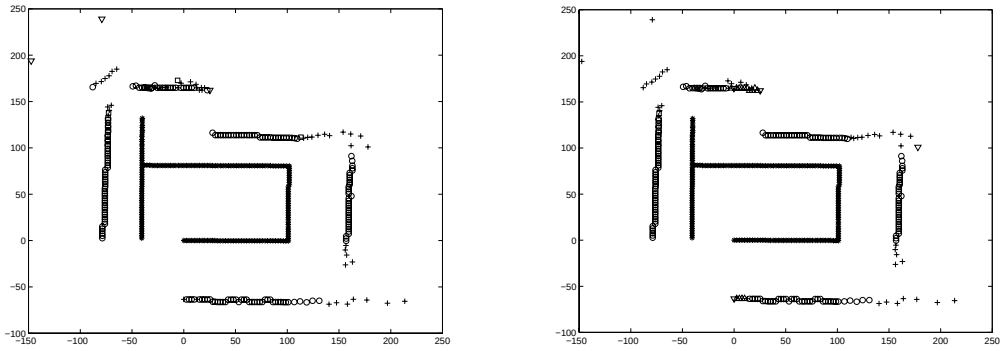
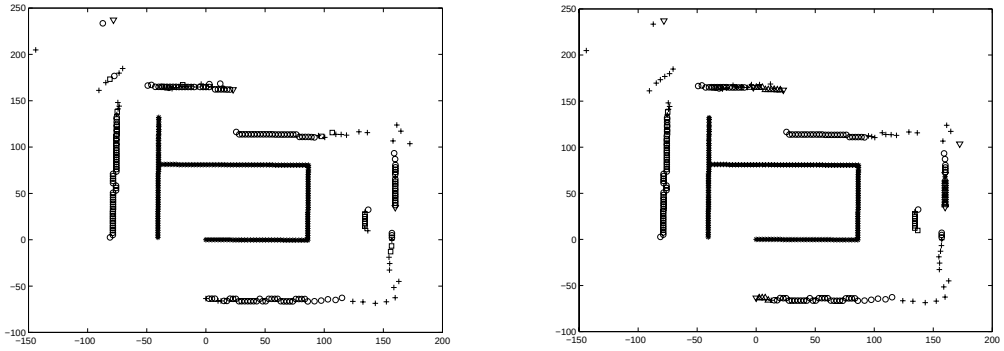


Figure 9.4: Infrared signals $x_r(n)$ and $x_l(n)$ which are collected by Nomad 200^{TM} (data set 12). These are employed in Algorithm II together with the sonar signals given in Figure 9.3.



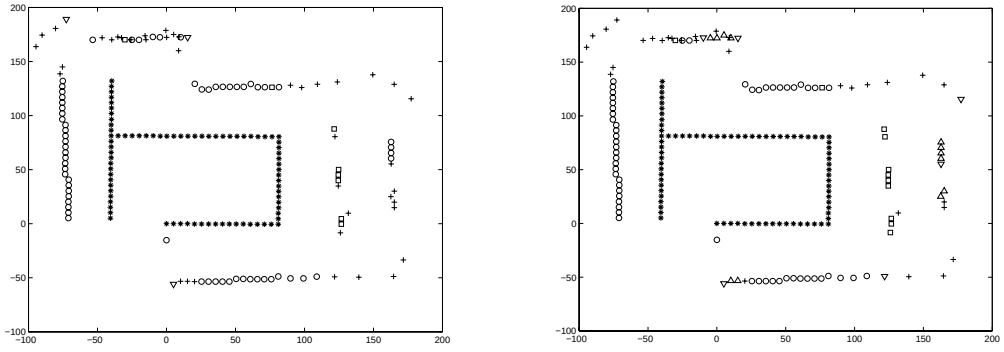
(a) (b)

Figure 9.5: Map which is extracted by employing (a) Algorithm I and (b) Algorithm II with data set 7. $*$: robot's position, $\circ$: plane, $+$: corner, $\square$: cylinder, $\bigtriangledown$: edge, and $\bigtriangleup$: unknown.



(a) (b)

Figure 9.6: Map which is extracted by employing (a) Algorithm I and (b) Algorithm II with data set 8. $*$: robot's position, $\circ$: plane, $+$: corner, $\square$: cylinder, $\bigtriangledown$: edge, and $\bigtriangleup$: unknown.



(a) (b)

Figure 9.7: Map which is extracted by employing (a) Algorithm I and (b) Algorithm II with data set 13. $*$: robot's position, $\circ$: plane, $+$: corner, $\square$: cylinder, $\bigtriangledown$: edge, and $\bigtriangleup$: unknown.

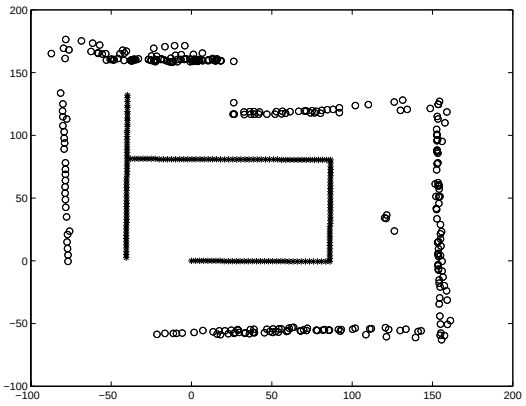


Figure 9.8: Laser readings ($\circ$) collected in data set 12 and the robot's position ($*$).

In this chapter, physically different sensors besides sonar are included for target classification. Two new algorithms are developed and applied to map-building with a mobile robot exploring its environment under the control of a human operator. In the next chapter, our concluding remarks are made and directions for future work are discussed.

Chapter 10

CONCLUSION

In this study, classification of target primitives which constitute the basic building blocks of typical uncluttered mobile robot environments has been considered. Sonar sensors placed at various vantage points in the environment make decisions about target type which are fused to reach a group decision through Dempster-Shafer evidential reasoning and majority voting. These sensors use both amplitude and TOF information in the sonar signals allowing for improved differentiation and localization.

Consistency problems arising in majority voting are addressed with a view to achieving high classification performance. This is done by introducing preference ordering among the possible target types and assigning reliability measures (which essentially serve as weights) to each decision-making node based on the target range and azimuth estimates it makes and the belief values it assigns to possible target types. Two different ways of preference ordering and five different reliability measure assignments have been considered. The effect of preference ordering on majority voting, and the effect of reliability measures on both fusion methods are tested experimentally. The results indicate that simple majority voting can provide fast and robust fusion in simple environments. However, when targets that cannot be classified by the

target differentiation algorithm are included in the environment, Dempster-Shafer method in its simple form can handle imprecise evidence more reliably than simple majority voting. When more sophisticated fusion methods incorporating reliability measures are employed, higher correct classification rates are obtained with preference-ordered majority voting than with evidential reasoning incorporating the same reliability measures. The overall performance of the various methods considered can be sorted in decreasing order as: majority voting with reliability measures and preference ordering, Dempster-Shafer method with reliability measures, Dempster-Shafer in its simple form, and simple majority voting.

Fuzzy c-means clustering algorithm and minimum description length principle to determine the suitable number of classes in sonar data collected from a number of classes of targets have also been employed. Most of the cases, $c = 7$ (which is the actual number of targets existing in the sonar data) gives the best number of classes in terms of compact and separate c-partitioning, minimum misclassification rate, and minimum cost.

In this thesis, various input signal representations, two different training algorithms, and different network structures have been considered for neural networks for improved target classification and localization with sonar. The input signals are different functional forms of amplitude and TOF patterns acquired by a real sonar system, and in most cases they are preprocessed before being used as inputs to the neural networks. The preprocessing techniques employed are discrete ordinary and fractional Fourier, Hartley and wavelet transforms and Kohonen's self-organizing feature map. Kohonen's self-organizing feature map is commonly used to extract the features of input data without supervision, resulting in scale-invariant classification. Here, it is used for feature extraction both prior to neural networks and also prior to a linear classifier. The performance of the different input signals are compared in terms of the successful classification and localization rates of the networks and their complexity. The training algorithms employed are back-propagation and generating-shrinking algorithms.

The networks trained with the generating-shrinking algorithm can only be used for determining the correct target type. Networks with modular structures have also been trained with the back-propagation algorithm for target classification and localization. When the results for non-modular and modular networks are compared, it is observed that the results for modular networks are in general slightly better than the results for non-modular ones. In most cases, the low-frequency component of the wavelet transform of the signal I_1 at resolution level $j = -2$ results in better classification and localization performance. For all input signals, the correct target differentiation rates of networks trained with the back-propagation and generating-shrinking algorithms are comparable except when the features obtained by using Kohonen's self-organizing feature map are used as input. In this case, the success rate obtained with using the generating-shrinking algorithm is much lower ($\leq 13\%$). Linear classifiers are also used to process the features extracted by Kohonen's self-organizing feature map and gave better results than processing the same features with neural networks. The maximum number of total neurons in the network layers is obtained with the input signals $\mathcal{F}^a(I_1)$ and I_1 for non-modular and modular network structures, respectively, the minimum number of total neurons in the network layers is obtained with the input signal $\mathrm{LFC}(\mathrm{DWT}(I_2))_2$ for both cases.

Statistical pattern recognition techniques which are two different interpretations of the k-nearest neighbor method (k-nearest neighbor and generalized k-nearest neighbor), kernel estimator, parametric density estimation with heteroscedastic and homoscedastic normal models, and linear discriminant analysis have also been included to classify the targets considered in this thesis. Better classification performance is achieved with non-parametric density estimation techniques (k-nearest neighbor methods and kernel estimator) than with parametric density estimation with normal models and linear discriminant analysis. Among the non-parametric methods considered, the best results are obtained by generalized k-nearest neighbor, followed by kernel estimator and k-nearest neighbor. Worst classification performance is obtained with linear discriminant

analysis, indicating that the different functional forms of amplitude and TOF patterns of the target primitives are not suitable for linear separation.

Although all of the methods considered in this thesis can be used for target differentiation, statistical pattern classification techniques and fuzzy c-means clustering algorithm cannot be used for target localization. The performances of all methods for target classification and localization are compared on three different test sets. These test sets include patterns acquired from targets situated at training locations as well as arbitrary locations, and targets which are not used for training and are somewhat different in size, shape, or roughness than those used for training. Target differentiation algorithm and fusion techniques (Dempster-Shafer evidential reasoning, simple majority voting, and voting fusion with preference ordering and reliability measures) employed in this thesis based on this algorithm can only be applied to plane, corner, and acute corner differentiation. However, all seven target types considered in the training phase and five extra target types added in the test phase can be differentiated using all other methods. Target localization performance of neural network classifiers are better than the target differentiation algorithm and fusion techniques employed based on this algorithm. Moreover, generalization capability of neural network classifiers, fuzzy c-means clustering algorithm, and target differentiation algorithm and fusion techniques are better than that of statistical pattern recognition techniques.

In this thesis, application of neural network classifiers (which is the *best* classification scheme resulting in the highest percentages in both target classification and localization) to map-building have been provided. A novel exploring strategy based on the generalized Voronoi diagram has been used.

The global maps of nine different test rooms are extracted experimentally by fusing the local maps of these rooms extracted at their meet points by our neural network classifier which employs $\mathrm{LFC}(\mathrm{DWT}(I_1))_2$ as input signal with different fusion schemes. These fusion schemes are Dempster-Shafer evidential reasoning (D-S), simple majority

voting (SMV), voting with preference ordering and reliability measures (VRM), and ordered voting fusion with preference ordering and reliability measures (k-VRM). Highest classification performances are obtained with k-VRM, and are followed by VRM, D-S, and SMV in the given order. A classification performance above 90% has been achieved for seven of these nine rooms. For rooms 7 and 8 which are the most complicated rooms in this study, these percentages are 81% and 73%, respectively.

Identifying the room the mobile robot is exploring has also been considered by using the meet point positions existing in this room as input to the neural networks. Extremely high performances are obtained in this case such that if the noise standard deviation is increased to ten times its value used in the training phase, the percentage of identifying the room correctly decreases only by 15%.

Infrared sensors besides sonars have also been employed in the target classification problem. Two new target classification algorithms have been developed. One of these algorithms employs only the sonar signals, the other one combines sonar and infrared signals to classify target types. In this case, all the cylinders are included in one class. The maps of Nomad 200^{TM} mobile robot's environments have been extracted experimentally by employing these two target classification algorithms. Processing both sonar and infrared signals together bring an improvement in target classification up to 5%.

The comparative analysis provided in this thesis are vital for robotics researchers searching for a method resulting in improved target classification and localization performance with sonar. While we have mostly concentrated on sonar sensors, the fusion techniques and classification schemes employed in this thesis can be useful in a wide variety of applications where multiple decision makers are involved.

Although an application example of the classification schemes employed in this thesis is provided in map-building of mobile robot environments, these classification schemes can find application in path-planning, obstacle avoidance, target-tracking, and robot

localization and navigation. For improving of the generalization capability of neural network classifiers, fusion of the output of several neural networks trained with the same input signal but with randomized training set from a common training pool can be investigated. Most of the researchers in this area are concentrated on the fusion of the output of several neural networks trained with the same input signal in the whole training pool for this purpose. There are various wavelet functions designed by using different design criterion in the literature [151–154]. The effect of these various wavelet functions on the generalization capability of neural network classifiers can also be investigated. Dempster's rule of combination is commutative and can only be used for the fusion of independent source of evidence. However, in some applications, fusion of the observations coming from the same sensor can be needed (i.e., dependent source of evidence) and the order of fusion can be important. For these kinds of applications, alternative combination rules can be investigated for sonar sensors in the light of the reliability measures assigned to them in this thesis.

Appendix A

PROGRAMS

Samples of the programs used throughout this thesis whose codes are given on the floppy disk at the back of the thesis are briefly introduced here.

There exists seven directories which are named as chapter2-6 and chapter8-9 in this floppy disk. In each directory, samples of the programs used in the corresponding chapters of the thesis are given (i.e., the programs used in Chapter 2 are in the directory chapter2, and so on.).

In directory chapter2, there exists one main C++ program belief.C, and six MATLAB programs which are fusion.m, vote.m, fusion-maxbel.m, vote-maxbel.m, fusion-dist.m, and vote-dist.m. Program belief.C assigns belief values to each target type which can be differentiated by target differentiation algorithm and estimates range and azimuth of a target, and angle of an acute corner. It also assigns belief values to the target range and azimuth estimates. Programs fusion.m and vote.m fuse the belief values assigned to each target type (plane, corner, acute corner) by each sensor node in the rectangular room having 15 sensor nodes by Dempster-Shafer evidential reasoning and simple majority voting, respectively. Programs fusion-maxbel.m and vote-maxbel.m fuse the belief values assigned to each target type by each sensor node based on maximum

belief criterion by Dempster-Shafer evidential reasoning and simple majority voting, respectively. Programs fusion-dist.m and vote-dist.m fuse the belief values assigned to each target type by each sensor node based on maximum or minimum distance criterion by Dempster-Shafer evidential reasoning and simple majority voting, respectively. These two programs use subprograms findy.m and findyr.m to find the sequence of sensor nodes in the fusion process based on the maximum and minimum distance criteria, respectively.

In directory chapter3, there exists two main programs which are diffvotestr-a.m and diffvotestr-b.m. Programs diffvotestr-a.m and diffvotestr-b.m fuse the belief values assigned to each target type (plane, corner, acute corner) by each sensor node by voting with preference ordering and reliability measures in the test rooms *Room A* and *B*, respectively.

In directory chapter4, there exists one main MATLAB program cmeans.m which finds the cluster centers and the fuzzy c-partition of the training set by employing fuzzy c-means clustering algorithm. It also find the fuzzy c-partition of test set using the cluster centers calculated in the training set. This program uses subprogram getvector.m given in this directory which produces required vector representations from training and test data sets.

In directory chapter5, there exist five main MATLAB programs which are getpatterns.m, gsa.m, ksofm.m, getoutput.m and getoutputmod.m and two PlaNet programs n.sty1 and n.sty1t. Program getpatterns.m produces input signals to the neural networks. It uses subprograms getinput.m for producing raw signals I_i $(i = 1, 2, 3)$, wave.m for obtaining discrete wavelet transform of the signals, findft.m for obtaining discrete ordinary and fractional Fourier, and Hartley transforms of the signals. Subprogram findft.m also uses another subprogram whose name is dFRT.m written by Çağatay Candan to compute the DFRT matrix. Program gsa.m finds the final weights of the network trained by the generating-shrinking algorithm and computes the output of the network for the given test set. This program uses subprograms gs.m for initial

weight assignment, shrink.m for shrinking phase of the generating-shrinking algorithm, getoutputs.m for obtaining output of the network for any test pattern, and step.m for calculating the output of known step function. Program ksofm.m extracts the features of the raw signals $I_i(i = 1, 2, 3)$ in the training set. It also finds weight matrix of a linear classifier which uses these features as input and calculates the output of this linear classifier for a given test set. Programs n.sty1 and n.sty1t calculate the weights and biases of the 3-layer non-modular and one module of modular neural networks using the back-propagation algorithm, respectively. Programs getoutput.m and getoutputmod.m compute the outputs of non-modular and modular neural networks trained by the back-propagation algorithm, respectively. They use subprogram activation.m to find the output of sigmoid function used in the neural networks trained by the back-propagation algorithm.

In directory chapter6, there exists six main MATLAB programs whose names are kNN.m, gkNN.m, kernel.m, parest-htr.m, parest-hm.m, and lda.m each one of which implements k-NN, generalized k-NN, kernel estimator, parameterized density estimator with heteroscedastic and homoscedastic normal models, and linear discriminant analysis, respectively. There exist three subprograms findneighboor.m, kerest.m, and getvector.m used in the MATLAB programs mentioned above. Subprogram findneighboor.m finds the nearest $k = 1$ to k_{max} neighbors of each test pattern in the training set and calculates the distances between each test pattern and its nearest neighbors. Subprogram kerest.m calculates kernel estimate of a pattern in the test set. Subprogram getvector.m produces vector representations.

In directory chapter8, there exists three main MATLAB programs which are getmapmp.m, fuserm.m, and getmapmg.m. Program getmapmp.m plots the local map of a given room at a given meet point by using the output of neural network classifier at this meet point. Program fuserm.m fuses the decisions of all classifiers at all meet points in a given room. It uses subprograms prompt.m to project range and azimuth estimates

of all classifiers at all meet points of a given room to the centroid of the room. Outputs of classifiers for type identification are also normalized to be used in the fusion process. This subprogram uses another subprogram prompt2av.m to make this projection operation to the output of classifier at a single scan angle. Program fuserm.m also uses another subprogram fusempt.m to fuse projected range and azimuth estimates and normalized outputs for target type (i.e., belief values). Program getmapmg.m plots the global map of a room by using the fused belief values of each target type, fused range and azimuth estimates obtained in this room. All three programs use subprogram pltrooms.m to plot surfaces existing in a given room and to find x and y coordinates of the centroid of this room.

In directory chapter9, there exists two MATLAB programs whose names are sonar.m and sonar-inf.m. Program sonar.m implements Algorithm I using the sonar signals collected by the Nomad 200^{TM} mobile robot. At the output, it plots sonar signals used in Algorithm I and the extracted map of the environment by employing Algorithm I. Program sonar-inf.m implements Algorithm II using both sonar and infrared signals collected by the Nomad 200^{TM} mobile robot. At the output, it plots sonar and infrared signals used in Algorithm II and the extracted map of the environment by employing Algorithm II.

Bibliography

[1] A. Elfes, "Sonar based real-world mapping and navigation," *IEEE Transactions on Robotics and Automation*, vol. RA-3, no. 3, pp. 249–265, June 1987.

[2] R. Kuc and B. Barshan, "Navigating vehicles through an unstructured environment with sonar," in *Proceedings of IEEE International Conference on Robotics and Automation*, Scottsdale, AZ, 14–19 May 1989, pp. 1422–1427.

[3] R. Kuc and B. V. Viard, "A physically-based navigation strategy for sonar-guided vehicles," *International Journal of Robotics Research*, vol. 10, no. 2, pp. 75–87, April 1991.

[4] R. Kuc and M. W. Siegel, "Physically-based simulation model for acoustic sensor robot navigation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 6, pp. 766–778, November 1987.

[5] A. Kurz, "Constructing maps for mobile robot navigation based on ultrasonic range data," *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, vol. 26, no. 2, pp. 233–242, April 1996.

[6] J. J. Leonard and H. F. Durrant-Whyte, *Directed Sonar Navigation*, Kluwer Academic Press, London, U.K., 1992.

[7] Ö. Bozma and R. Kuc, "A physical model-based analysis of heterogeneous environments using sonar—ENDURA method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 5, pp. 497–506, May 1994.

[8] R. Kuc, "Three-dimensional tracking using qualitative bionic sonar," *Robotics and Autonomous Systems*, vol. 11, no. 2, pp. 213–219, 1993.

[9] J. Borenstein and Y. Koren, "Obstacle avoidance with ultrasonic sensors," *IEEE Transactions on Robotics and Automation*, vol. RA-4, no. 2, pp. 213–218, April 1988.

[10] H. Peremans, K. Audenaert, and J. M. Van Campenhout, "A high-resolution sensor based on tri-aural perception," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 1, pp. 36–48, February 1993.

[11] L. Kleeman and R. Kuc, "Mobile robot sonar for target localization and classification," *International Journal of Robotics Research*, vol. 14, no. 4, pp. 295–318, August 1995.

[12] B. Barshan and B. Ayrulu, "Performance comparison of four methods of time-of-flight estimation for sonar waveforms," *Electronics Letters*, vol. 34, no. 16, pp. 1616–1617, 6 August 1998.

[13] Polaroid Corporation, Ultrasonic Components Group, 119 Windsor St., Cambridge, MA, *Polaroid Manual*, 1997.

[14] Ö. Bozma and R. Kuc, "Building a sonar map in a specular environment using a single mobile sensor," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 12, pp. 1260–1269, December 1991.

[15] Ö. Bozma and R. Kuc, "Characterizing pulses reflected from rough surfaces using ultrasound," *The Journal of the Acoustical Society of America*, vol. 89, no. 6, pp. 2519–2531, June 1991.

[16] A. M. Sabatini, "Statistical estimation algorithms for ultrasonic detection of surface features," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Munich, Germany, 12–16 September 1994, pp. 1845–1852.

[17] B. Barshan and A. Ş. Sekmen, "Radius of curvature estimation and localization of targets using multiple sonar sensors," *The Journal of the Acoustical Society of America*, vol. 105, no. 4, pp. 2318–2331, April 1999.

[18] M. L. Hong and L. Kleeman, "Ultrasonic classification and location of 3-D room features using maximum likelihood estimation I," *Robotica*, vol. 15, no. Part 5, pp. 483–491, September/October 1997.

[19] M. L. Hong and L. Kleeman, "Ultrasonic classification and location of 3-D room features using maximum likelihood estimation II," *Robotica*, vol. 15, no. Part 6, pp. 645–652, November/December 1997.

[20] J. Manyika and H. F. Durrant-Whyte, *Data Fusion and Sensor Management: A Decentralized Information-Theoretic Approach*, Ellis Horwood, New York, NY, 1994.

[21] B. Ayrulu and B. Barshan, "Identification of target primitives with multiple decision-making sonars using evidential reasoning," *International Journal of Robotics Research*, vol. 17, no. 6, pp. 598–623, June 1998.

[22] B. Barshan and R. Kuc, "Differentiating sonar reflections from corners and planes by employing an intelligent sensor," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 6, pp. 560–569, June 1990.

[23] R. Kuc, "Biomimetic sonar recognizes objects using binaural information," *The Journal of the Acoustical Society of America*, vol. 102, no. 2, pp. 689–696, August 1997.

[24] B. Barshan, B. Ayrulu, and S. W. Utete, "Neural network based target differentiation using sonar for robotics applications," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 4, pp. 435–442, August 2000.

[25] R. P. Lippman, "An introduction to computing with neural nets," *IEEE ASSP Magazine*, pp. 4–22, April 1987.

[26] B. Widrow and R. Winter, "Neural nets for adaptive filtering and adaptive pattern recognition," *IEEE Computer*, vol. 21, no. 3, pp. 25–40, March 1988.

[27] W. Chang, B. Bosworth, and G. C. Carter, "Results of using an artificial neural network to distinguish single echoes from multiple sonar echoes," *The Journal of the Acoustical Society of America*, vol. 94, no. 3, pp. 1404–1408, September 1993, Part 1.

[28] C. Prieve and D. Marchette, "An application of neural networks to a data fusion problem," *1987 Tri-Service Data Fusion Technical Proceedings, John Hopkins University*, pp. 226–236, June Baltimore, MD, 1987.

[29] H. L. Roitblat, W. W. L. Au, P. E. Nachtigall, R. Shizumura, and G. Moons, "Sonar recognition of targets embedded in sediment," *Neural Networks*, vol. 8, no. 7/8, pp. 1263–1273, 1995.

[30] J. A. Simmons, P. A. Saillant, J. M. Wotton, T. Haresign, M. J. Ferragamo and C. F. Moss, "Composition of biosonar images for target recognition by echolocating bats," *Neural Networks*, vol. 8, no. 7/8, pp. 1239–1261, 1995.

[31] W. W. L. Au, "Comparison of sonar discrimination—dolphin and artificial neural network," *The Journal of the Acoustical Society of America*, vol. 95, no. 5, pp. 2728–2735, May 1994, Part 1.

[32] G. B. Willson, "Radar classification using a neural network," in *Optical Engineering and Photonics in Aerospace Sensing: Application of Neural Networks, Proceedings of SPIE*, April 1990, vol. 1294, pp. 200–210.

[33] S. Watanabe and M. Yoneyama, "An ultrasonic visual sensor for three-dimensional object recognition using neural networks," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 2, pp. 240–249, April 1992.

[34] I. E. Dror, M. Zagaeski, and C. F. Moss, "3-dimensional target recognition via sonar-a neural network model," *Neural Networks*, vol. 8, no. 1, pp. 149–160, January 1995.

[35] R. P. Gorman and T. J. Sejnowski, "Learned classification of sonar targets using a massively parallel network," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 7, pp. 1135–1140, July 1998.

[36] T. Ogawa, K. Kameyama, R. Kuc, and Y. Kosugi, "Source localization with network inversion using an answer-in-weights scheme," *IEICE Transactions on Information and Systems*, vol. E79-D, no. 5, pp. 608–619, May 1996.

[37] R. K. Miller and T. C. Walker, *Neural Network Applications and Products*, vol. 1, SEAI Technical Publications, Norcross, GA, 1992.

[38] E. Alpaydın, "Multiple networks for function learning," in *Proceedings of IEEE International Conference on Neural Networks*, San Francisco, March 1993, pp. 9–14.

[39] W. W. L. Au, L. N. Andersen, A. R. Rasmussen, H. L. Roitblat, and P. E. Nachtigall, "Neural network modeling of a dolphin's sonar discrimination capabilities," *The Journal of the Acoustical Society of America*, vol. 98, no. 1, pp. 43–50, July 1995.

[40] G. Shafer, *A Mathematical Theory of Evidence*, Princeton University Press, Princeton, NJ, 1976.

[41] J.-B. Yang and M. G. Singh, "An evidential reasoning approach for multiple-attribute decision making with uncertainty," *IEEE Transactions on Systems Man and Cybernetics*, vol. 24, no. 1, pp. 1–18, January 1994.

[42] P. Krause and D. Clark, *Representing Uncertain Knowledge: an Artificial Intelligence Approach*, Intellect Books, Bristol, U.K., 1993.

[43] B. Ayrulu, B. Barshan, and S. W. Utete, "Target identification with multiple logical sonars using evidential reasoning and simple majority voting," in *Proceedings of IEEE International Conference on Robotics and Automation*, Albuquerque, NM, 20–25 April 1997, pp. 2063–2068.

[44] S. W. Utete, B. Barshan, and B. Ayrulu, "Voting as validation in robot programming," *International Journal of Robotics Research*, vol. 18, no. 4, pp. 401–413, April 1999.

[45] H. Choset and J. Burdick, "Sensor based planning, Part I: the generalized Voronoi graph," in *Proceedings of IEEE International Conference on Robotics and Automation*, Nagoya, Japan, 21–27 May 1995.

[46] H. Choset and J. Burdick, "Sensor based planing, Part II: incremental construction of the generalized Voronoi graph," in *Proceedings of IEEE International Conference on Robotics and Automation*, Nagoya, Japan, 21–27 May 1995.

[47] H. Choset, I. Konukseven, and J. Burdick, "Mobile robot navigation: issues in implementating the generalized Voronoi graph in the plane," in *Proceedings IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Washington D.C., 8–11 December 1996, pp. 241–248.

[48] B. Ayrulu, "Classification of Target Primitives with Sonar using two Non-parametric Data-Fusion Methods," M.S. thesis, Bilkent University, Department of Electrical Engineering, Ankara, Turkey, July 1996.

[49] P. Hauptmann, *Sensors: Principles and Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1993.

[50] D. Başkent and B. Barshan, "Surface profile determination from multiple sonar data using morphological processing," *International Journal of Robotics Research*, vol. 18, no. 8, pp. 788–808, August 1999.

[51] J. Zemanek, "Beam behavior within the nearfield of a vibrating piston," *The Journal of the Acoustical Society of America*, vol. 49, no. 1 (Part 2), pp. 181–191, January 1971.

[52] B. Barshan, *A Sonar-Based Mobile Robot for Bat-Like Prey Capture*, Ph.D. thesis, Yale University, Department of Electrical Engineering, New Haven, CT, December 1991.

[53] M. K. Brown, "The extraction of curved surface features with generic range sensors," *International Journal of Robotics Research*, vol. 5, no. 1, pp. 3–18, Spring 1986.

[54] B. Barshan, "Location and curvature estimation of spherical targets using a flexible sonar configuration," in *Proceedings of IEEE International Conference on Robotics and Automation*, Minneapolis, MN, 22–28 April 1996, pp. 1218–1223.

[55] B. Ayrulu, B. Barshan, I. Erkmen, and A. Erkmen, "Evidential logical sensing using multiple sonars for the identification of target primitives in a mobile robot's environment," in *Proceedings IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Washington D.C., 8–11 December 1996, pp. 365–372.

[56] A. M. Erkmen and H. E. Stephanou, "Information fractals for evidential pattern classification," *IEEE Transactions on Systems Man and Cybernetics*, vol. 20, no. 5, pp. 1103–1114, September/October 1990.

[57] H. Y. Hau and R. L. Kashyap, "Belief combination and propagation in a lattice-structured inference network," *IEEE Transactions on Systems Man and Cybernetics*, vol. 20, no. 1, pp. 45–58, January/February 1990.

[58] R. R. Murphy, "Adaptive rule of combination for observations over time," in *Proceedings IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Washington D.C., 8–11 December 1996, pp. 125–131.

[59] D. Fixsen and R. P. S. Mahler, "The modified Dempster-Shafer approach to classification," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 27, no. 1, pp. 96–104, January 1997.

[60] T. Denoeux, "A k-nearest neighborhood classification rule based on Dempster-Shafer theory," *IEEE Transactions on Systems Man and Cybernetics*, vol. 25, no. 5, pp. 804–813, May 1995.

[61] A. P. Tirumalai, B. G. Schunck, and R. C. Jain, "Evidential reasoning for building environment maps," *IEEE Transactions on Systems Man and Cybernetics*, vol. 25, no. 1, pp. 10–20, January 1995.

[62] D. Pagac, E. M. Nebot, and H. F. Durrant-Whyte, "An evidential approach to map-building for autonomous vehicles," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 4, pp. 623–629, August 1998.

[63] R. R. Murphy, "Dempster-Shafer theory for sensor fusion in autonomous mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 2, pp. 197–206, April 1998.

[64] J. J. Gertler and K. C. Anderson, "An evidential reasoning extension to qualitative model-based failure diagnosis," *IEEE Transactions on Systems Man and Cybernetics*, vol. 22, no. 2, pp. 275–288, March/April 1992.

[65] D. M. Buede and P. Girardi, "A target identification comparison of Bayesian and Dempster-Shafer multisensor fusion," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 27, no. 5, pp. 569–577, September 1997.

[66] B. Parhami, "Voting algorithms," *IEEE Transactions on Reliability*, vol. 43, no. 4, pp. 617–629, December 1994.

[67] J. Mao, P. J. Flynn, and A. K. Jain, "Integration of multiple feature groups and multiple views into a 3D object recognition system," *Computer Vision and Image Understanding*, vol. 62, no. 3, pp. 309–325, November 1995.

[68] L. A. Klein, *Sensor and Data Fusion Concepts and Applications*, SPIE Optical Engineering Press, Volume TT 14 (Tutorial Texts in Optical Engineering), Bellingham, WA, 1993, Section on Voting Fusion, pp.73–90.

[69] L. Lam and C. Y. Suen, "Application of majority voting to pattern recognition: an analysis of its behavior and performance," *IEEE Transactions on Systems Man and Cybernetics*, vol. 27, no. 5, pp. 553–568, September 1997.

[70] K. J. Rosenblatt, "DAMN: A distributed architecture for mobile navigation," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 9, no. 2–3, pp. 339–360, 1997.

[71] P. Pirjanian, J. A. Fayman, and H. I. Christensen, "Improving task reliability by fusion of redundant homogeneous modules using voting schemes," in *Proceedings of IEEE International Conference on Robotics and Automation*, Albuquerque, NM, 20–25 April 1997, pp. 425–430.

[72] Y.-W. Leung, "Maximum likelihood voting for fault-tolerant software with finite output-space," *IEEE Transactions on Reliability*, vol. 44, no. 3, pp. 419–427, September 1995.

[73] K. J. Arrow, *Social Choice and Individual Values*, New York: Wiley, 1951.

[74] Panasonic Corporation, "Ultrasonic ceramic microphones," 12 Blanchard Road, Burlington, MA, 1989.

[75] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, Wiley Interscience, New York, NY, 1973.

[76] P. Turney, "Theoretical analysis of cross-validation error and voting in instance-based learning," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 6, no. 4, pp. 331–360, October-December 1994.

[77] P. Turney, "A theory of cross-validation error," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 6, no. 4, pp. 361–391, October-December 1994.

[78] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *Journal of Cybernetics*, vol. 3, pp. 32–57, 1974.

[79] J. C. Bezdek, *Fuzzy Mathematics in Pattern Classification*, Ph.D. thesis, Cornell University, Department of Applied Mathematics, Ithaca, NY, 1973.

[80] X. L. Xie and G. Beni, "A validity measure for fuzzy clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-13, no. 8, pp. 841–847, August 1991.

[81] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, pp. 465–478, 1978.

[82] J. Rissanen, "Universal coding, information, prediction, and estimation," *IEEE Transactions on Information Theory*, vol. 30, pp. 629–636, 1984.

[83] J. Rissanen, "Minimum-description-length principle," *Encyclopedia of Statistic Sciences*, vol. 5, pp. 523–527, 1987.

[84] Y. G. Leclerc, "Constructing simple stable description for image partitioning," *International Journal of Computer Vision*, vol. 3, pp. 73–102, 1989.

[85] A. P. Pentland, "Part segmentation for object recognition," *Neural Computation*, vol. 1, pp. 82–91, 1999.

[86] A. Leonardis, A. Gupta, and R. Bujcsy, "Segmantation of range images as the search for geometric parametric models," *International Journal of Computer Vision*, vol. 14, pp. 253–277, 1995.

[87] R. Joshi and A. C. Sanderson, "Minimal representation multisensor fusion using differential evolution," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 29, no. 1, pp. 63–76, January 1999.

[88] A. L. Oliveira and A. S. Vincentelli, "Using the minimum description length principle to inferreduced ordered decision graphs," *Machine Learning*, vol. 25, pp. 23–50, 1996.

[89] H. Bischof and A. Leonardis, "Finding optimal neural networks for land use classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 36, no. 1, pp. 337–341, January 1998.

[90] X. M. Gao, S. J. Ovaska, M. Lethokangas, and J. Saarinen, "Modeling of speech signals using an optimal neural network structure based on pmdl principle," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 2, pp. 177–180, March 1998.

[91] J. Rissanen, "Fisher information and stochastic complexity," *IEEE Transactions on Information Theory*, vol. 42, pp. 40–47, 1996.

[92] P. Fua and A. J. Hanson, "Objective functions for feature discrimination," in *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, Detroit, MI, 1989, pp. 1596–1602.

[93] B. C. Bai and N. H. Farhat, "Learning networks for extrapolation and radar target identification," *Neural Networks*, vol. 5, no. 3, pp. 507–529, 1992.

[94] M. Cohen, H. Franco, N. Morgan, D. Rumelhart, and V. Abrash, "Context-dependent multiple distribution phonetic modelling with MLPs," in *Advances in Neural Information Processing Systems*, S. J. Hanson, J. D. Cowan, and C. L. Giles, Ed., pp. 649–657. Morgan Kaufmann, San Mateo, CA, 1993.

[95] K. S. Narendra and K. Parthasarathy, "Gradient methods for the optimization of dynamic systems containing neural networks," *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 252–262, March 1991.

[96] M. I. Jordan and R. A. Jacobs, "Learning to control an unstable system with forward modeling," in *Advances in Neural Information Processing Systems 2*, D. S. Touretzky, Ed., pp. 324–331. Morgan Kaufmann, San Mateo, CA, 1990.

[97] M. Galicki, H. Witte, J. Dörschel, M. Eiselt, and G. Griessbach, "Common optimization of adaptive processing units and a neural network during the learning period: application in EEG pattern recognition," *Neural Networks*, vol. 10, no. 6, pp. 1153–1163, August 1997.

[98] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in Neural Information Processing Systems 2*, D. S. Touretzky, Ed., pp. 396–404. Morgan Kaufmann, San Mateo, CA, 1990.

[99] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall, New Jersey, NJ, 1994.

[100] D. W. Ruck, S. K. Rogers, M. Kabrisky, M. E. Oxley, and B. W. Suter, "The multilayer perceptron as an approximation to a Bayes optimal discriminant function," *IEEE Transactions on Neural Networks*, vol. 1, no. 4, pp. 296–298, December 1990.

[101] R. Anand, K. Mehrotra, C. K. Mohan, and S. Ranka, "Efficient classification for multiclass problems using modular neural networks," *IEEE Transactions on Neural Networks*, vol. 6, no. 1, pp. 117–124, January 1995.

[102] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, October 1990.

[103] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagation errors," *Nature*, vol. 323, pp. 533–536, 1986, London.

[104] Y. Q. Chen, D. W. Thomas, M. S. Nixon, "Generating-shrinking algorithm for learning arbitrary classification," *Neural Networks*, vol. 7, no. 9, pp. 1477–1489, 1994.

[105] R. N. Bracewell, *The Fourier Transform and Its Applications*, McGraw-Hill, New York, NY, 1986.

[106] H. M. Ozaktas, M. A. Kutay, and D. Mendlovic, "Introduction to the fractional Fourier transform and its applications," in *Advances in Imaging and Electron Physics*, P. W. Hawkes, Ed., vol. 106, pp. 239–291. Academic Press, San Diego, CA, 1999.

[107] H. M. Ozaktas, Z. Zalevsky, and M. A. Kutay, *The Fractional Fourier Transform with Applications in Optics and Signal Processing*, John Wiley & Sons, New York, NY, 2000.

[108] D. Mendlovic and H. M. Ozaktas, "Fourier transforms and their optical implementation: I," *The Journal of the Optical Society of America A*, vol. 10, pp. 1875–1881, 1993.

[109] H. M. Ozaktas and D. Mendlovic, "Fourier transforms of fractional order and their optical interpretation," *Optics Communications*, vol. 101, pp. 163–169, 1993.

[110] H. M. Ozaktas and D. Mendlovic, "Fractional Fourier transforms and their optical implementation: II," *The Journal of the Optical Society of America A*, vol. 10, pp. 2522–2531, 1993.

[111] H. M. Ozaktas and D. Mendlovic, "Fractional Fourier Optics," *The Journal of the Optical Society of America A*, vol. 12, pp. 743–751, 1995.

[112] L. B. Almeida, "The fractional Fourier transform and time-frequency representations," *IEEE Transactions on Signal Processing*, vol. 42, no. 11, pp. 3084–3091, November 1994.

[113] M. A. Kutay, M. F. Erden, H. M. Ozaktas, O. Arıkan, Ö. Güleryüz, and Ç. Candan, "Space-bandwidth-efficient realizations of linear systems," *Optics Letters*, vol. 23, pp. 1069–1071, 1998.

[114] D. Mendlovic, Z. Zalevsky, and H. M. Ozaktas, "Applications of the fractional Fourier transform to optical pattern recognition," in *Optical Pattern Recognition*, F. T. S. Yu and S. Jutamulia, Ed., chapter 4, pp. 89–125. Cambridge University Press, Cambridge, 1998.

[115] M. A. Kutay, H. M. Ozaktas, O. Arıkan, and L. Onural, "Optimal filtering in fractional Fourier domains," *IEEE Transactions on Signal Processing*, vol. 45, pp. 1129–1143, 1997.

[116] M. A. Kutay, H. Özaktaş, H. M. Ozaktas, and O. Arıkan, "The fractional Fourier domain decomposition," *Signal Processing*, vol. 77, pp. 105–109, 1999.

[117] H. M. Ozaktas, B. Barshan, D. Mendlovic, and L. Onural, "Convolution, filtering, and multiplexing in fractional Fourier domains and their relation to chirp and wavelet transforms," *The Journal of the Optical Society of America A*, vol. 11, no. 2, pp. 547–559, February 1994.

[118] H. M. Ozaktas, B. Barshan, and D. Mendlovic, "Convolution and filtering in fractional Fourier domains," *Optical Review*, vol. 1, pp. 15–16, 1994.

[119] B. Barshan, M. A. Kutay, and H. M. Ozaktas, "Optimal filtering with linear canonical transformations," *Optics Communications*, vol. 135, no. 1–3, pp. 32–36, February 1997.

[120] M. A. Kutay and H. M. Ozaktas, "Optimal image restoration with the fractional Fourier transform," *The Journal of the Optical Society of America A*, vol. 15, pp. 825–833, 1998.

[121] İ. Ş. Yetik, H. M. Ozaktas, B. Barshan, and L. Onural, "Perspective projections in the space-frequency plane and fractional Fourier transforms," *The Journal of the Optical Society of America A*, vol. 17, no. 12, pp. 2382–2390, December 2000.

[122] A. C. McBride and F. H. Kerr, "On Namias's fractional Fourier transform," *IMA Journal of Applied Mathematics*, vol. 39, pp. 159–175, 1987.

[123] S. Cohen, *Time-Frequency Analysis*, Prentice Hall, New Jersey, NJ, 1995.

[124] H. M. Ozaktas, O. Arıkan, M. A. Kutay, and G. Bozdağı, "Digital computation of the fractional Fourier transform," *IEEE Transactions on Signal Processing*, vol. 44, pp. 2141–2150, 1996.

[125] Ç. Candan, "The Discrete Fractional Fourier Transform," M.S. thesis, Bilkent University, Department of Electrical Engineering, Ankara, Turkey, July 1998.

[126] Ç. Candan, M. A. Kutay, and H. M. Ozaktas, "The discrete fractional Fourier transform," *IEEE Transactions on Signal Processing*, vol. 48, pp. 1329–1337, 2000.

[127] R. N. Bracewell, *The Hartley Transform*, Oxford University Press, New York, NY, 1986.

[128] R. C. Gonzalez and P. Wintz, *Digital Image Processing*, Addison-Wesley, 1987.

[129] D. F. Marshall, W. K. Jenkins, and J. J. Murphy, "The use of orthogonal transforms for improving performance of adaptive filters," *IEEE Transactions on Circuits and Systems*, vol. 36, pp. 474–484, April 1989.

[130] C. K. Chui, *An Introduction to Wavelets*, Academic Press, San Diego, CA, 1992.

[131] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, no. 1, pp. 59–69, 1982.

[132] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, July 1989.

[133] B. Barshan and B. Ayrulu, "Fractional Fourier transform preprocessing for neural networks and its application to object recognition," submitted to *Neural Networks*, April 2001.

[134] M. Rosenblatt, "Remarks on some non-parametric estimates of a density function," *Annals of Mathematical Statistics*, vol. 27, pp. 832–837, 1956.

[135] E. Fix and J. L. Hodges, "Discriminatory analysis, nonparametric discrimination, consistency properties," Tech. Rep. 21-49-004 4, United States Air Force, School of Aviation Medicine, Randolph Field, Texas, 1951.

[136] E. Parzen, "On estimation of a probability density function and mode," *Annals of Mathematical Statistics*, vol. 33, pp. 1065–1076, 1962.

[137] R. Cacoullos, "Estimation of multivariate density," *Annals of the Institute of Statistical Mathematics (Tokyo)*, vol. 18, no. 2, pp. 179–189, 1966.

[138] D. J. Hand, *Discrimination and Classification*, John Wiley & Sons, New York, NY, 1986.

[139] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman and Hall, New York, NY, 1986.

[140] D. J. Hand, *Kernel Discriminant Analysis*, Research Studies Press, New York, 1982.

[141] M. P. Wand and M. C. Jones, *Kernel Smoothing*, Chapman and Hall, London, 1995.

[142] S. J. Sheather, "The performance of six popular bandwidth selection methods on some real data sets," *Computational Statistics*, vol. 7, pp. 225–250, 1992.

[143] V. K. Rohatgi, *An Introduction to Probability Theory and Mathematical Statistics*, John Wiley & Sons, New York, NY, 1976.

[144] G. J. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition*, John Wiley & Sons, New York, NY, 1992.

[145] B. Ayrulu and B. Barshan, "Reliability measure assignment to sonar for robust target diffentiation," *Pattern Recognition*, accepted May 2001.

[146] B. Ayrulu and B. Barshan, "Statistical pattern recognition techniques and linear discriminant analysis for target classification with sonar," Tech. Rep., Bilkent University, Ankara, Turkey, 2000.

[147] B. Ayrulu and B. Barshan, "Fuzzy clustering and minimum description length principle for determining the number of classes of targets with sonar," Tech. Rep., Bilkent University, Ankara, Turkey, 2000.

[148] B. Ayrulu and B. Barshan, "Neural networks for improved target differentiation and localization with sonar," *Neural Networks*, vol. 14, no. 3, pp. 355–373, April 2001.

[149] B. Ayrulu and B. Barshan, "Comparative analysis of different approaches to target differentiation and localization with sonar," Tech. Rep., Bilkent University, Ankara, Turkey, 2000.

[150] C. O. Dunlaing and C. K. Yap, "A retraction method for planning the motion of a disc," *Algorithmica*, vol. 6, pp. 104–111, 1985.

[151] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Commun. Pure Appl. Math.*, vol. 41, pp. 909–996, 1988.

[152] C. Herley and M. Vetterli, "Biorthogonal bases of symmatric compactly supported wavelets," in *Wavelets, Fractals, and Fourier Transforms*, M. Farge, J. C. R. Hunt, and J. C. Vassilica, Ed., pp. 91–108. Oxford University Press, Oxford, 1993.

[153] C. Herley and M. Vetterli, "Linear phase wavelets: Theory and design," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toronto, Canada, May 1991, pp. 2017–2020.

[154] P. P. Vaidyanathan and P. -Q. Hoang, "Lattice structures for optimal design and robust implementation of two-band perfect reconstruction QMF banks," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, pp. 81–94, January 1988.

Vita

Birsel Ayrulu (Erdem) was born in Konya, Turkey, on 26 November, 1972. She received the B.S. degree in electrical engineering from Middle East Technical University (METU) and the M.S. degree in electrical engineering from Bilkent University, Ankara, Turkey in 1994 and 1996, respectively. She was ranked first in the department and second in the university while graduating from METU. She received Assoc. Prof. Dr. Bülent Kerim Altay Award three times in 1992, 1993, and 1994. She then pursued her Ph. D. studies at the Department of Electrical and Electronics Engineering at Bilkent University. During her M.S. and Ph. D. studies, she was a research and teaching assistant at the same department, and a scholar of the BDP program of TÜBİTAK. She is the recipient of the First Alper Atalay Best Student Paper Award in the *9th IEEE Conference on Signal Processing and Applications, 2001.* Her current research interests include intelligent sensing systems, sonar sensing, sensor data fusion, learning methods, target differentiation and localization, sensor-based robotics, and digital signal processing.