A STUDY OF DEMPSTER-SHAFER’S THEORY OF EVIDENCE IN COMPARISON TO CLASSICAL PROBABILITY COMBINATION

A Thesis presented to the
Electrical Engineering Faculty of
California Polytechnic State University, San Luis Obispo

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Electrical Engineering

by
Scott J. Seims
June 2009
COMMITTEE MEMBERSHIP

TITLE: A study of Dempster-Shafer’s Theory of Evidence in Comparison to Classical Probability Combination

AUTHOR: Scott J. Seims

DATE SUBMITTED: May 29, 2009

COMMITTEE CHAIR: John Saghri

COMMITTEE MEMBER: Wayne C. Pilkington

COMMITTEE MEMBER: Xiao-Hua Yu
ABSTRACT
A Study of Dempster-Shafer’s Theory of Evidence in Comparison to Classical Probability Combination

Scott J. Seims

This thesis is an assessment on the effectiveness of Dempster-Shafer’s Theory of Evidence in comparison to Classical Probabilistic Combination as it applies to Synthetic Aperture Radar (SAR) Automatic Target Recognition (ATR). A three feature based classifier (peaks, corners and edges) ATR system is presented. These classifiers are assumed to be independent. The results of both the weight-based Maximum Likelihood and Dempster-Shafer’s Theory of Evidence data fusion techniques are presented.

Using Dempster-Shafer an accuracy of 77.50 percent is obtained, which is less than the 86.25 percent accuracy of target-specific weight-based Maximum Likelihood. Inagaki’s Unified Combination Rule is implemented as a means to increase SAR ATR accuracy and explore further modifications of Dempster-Shafer. Inagaki’s Unified Combination Rule contains Yager’s and Dempster’s Combination Rules as well as Inagaki’s Extra Rule. The maximum accuracy achieved using Inagaki’s Unified Combination Rule was 75.00 percent.

It was concluded that this application lends itself better to classical probabilistic combination. Due to the single sensor (SAR) and the quasi-independence of the three feature based classifiers, Dempster-Shafer’s Theory of Evidence can not be utilized to its full potential.

This thesis is a continuation of Hausdorff Probabilistic Feature Analysis in SAR Image Recognition by Chessa Guillas. Thesis research is directed by Dr. John A. Saghri and sponsored by Raytheon Space & Airborne Systems, El Segundo, California.
ACKNOWLEDGMENTS

First and foremost I have to thank my parents, Gary and Marcie Seims for encouraging me through and, debatably more important, funding my education. Without their support none of this would have been possible.

I would like to take this opportunity to thank Professor Saghri for giving me the chance to work and research an amazing topic, none of which would have been possible without the support of Raytheon. I would also like to thank Professor Pilkington and Professor Yu for taking part in my defense as committee members as well as for all advice I received.

While that covers all those directly responsible for the completion of this thesis there are many others that have helped me to arrive at this moment. I would like to thank my brother, Jeremy Seims, for all motivational phone calls and my friends for not abandoning me while I focused on thesis and lost track of the outside world.

Thank you to every Professor who has supplied me with the necessary knowledge and tools to succeed in Electrical Engineering.

Sincerely,

Scott Seims
# TABLE OF CONTENTS

## LIST OF TABLES

| IX |

## LIST OF FIGURES

| X |

## 1. INTRODUCTION

1.1 DOCUMENT OVERVIEW 2
1.2 SOFTWARE PLATFORM 2

## 2. BACKGROUND

2.1 SYNTHETIC APERTURE RADAR 3
2.1.2 BRIEF HISTORIC PERSPECTIVE 3
2.1.3 SAR BASICS 4
2.1.4 SAR GEOMETRY 5
2.1.5 BRIEF GENERIC SAR EQUATION OVERVIEW 6
2.2 DATA DEVELOPMENT 8
2.2.1 BASE IMAGES 9
2.2.2 PREPROCESSING 9
2.2.2.1 Median Filtering 9
2.2.2.2 Gaussian Filtering 10
2.3 FEATURE EXTRACTION 10
2.3.1 EDGES 10
2.3.2 CORNERS 11
2.3.3 PEAKS 11
2.4 HAUSDORFF DISTANCE 12
2.4.1 DISTANCE TRANSFORM 12
2.4.2 AVERAGE HAUSDORFF DISTANCE AND PARTIAL HAUSDORFF DISTANCE 13
2.4.3 TRAINING SET DEVELOPMENT 13
2.4.4 CLASSIFICATION 14

## 3. DEMPSTER-SHAFER’S THEORY OF EVIDENCE 22

3.1 INTRODUCTION TO DEMPSTER-SHAFER’S THEORY OF EVIDENCE 22
3.2 BASIC PROBABILITY ASSIGNMENTS 24
3.3 BELIEF FUNCTIONS 26
3.4 BELIEF INTERVALS 27
3.5 COMBINATION 28
3.6 DANGERS OF DEMPSTER-SHAFER 29
### 4. YAGAR’S MODIFIED DEMPSTER-SHAFER

4.1 Introduction

4.2 Quasi-Associative Operators

4.2.1 Quasi-Associative Framework Applied to Dempster’s Combination Rule

4.3 Yager’s Modified Dempster Shafer

### 5. INAGAKI’S UNIFIED COMBINATION RULE

5.1 Introduction

5.2 Unified Combination Rule

5.3 Inagaki’s Extra Rule

### 6. IMPLEMENTATION

6.1 Phase One: Proof of Concept

6.2 Phase Two: Generalization of Phase One

6.3 Phase Three: Inagaki’s Unified Combination Rule

### 7. RESULTS

7.1 Phase One and Two

7.2 Phase Three

7.3 Combination Rules Compared to Chessa Guilas’ Results

### 8. CONCLUSIONS AND FUTURE WORK

8.1 Dempster-Shafer as Compared to Results Found by Chessa Guilas

8.2 Reasons for the Results of Different Combination Rules

8.2.1 Lack of Results for Yager’s Combination Rule

8.2.2 Inagaki’s Unified Combination Rule

8.3 Future Work

### REFERENCES:

### APPENDIX A: ALGORITHM CODE

**Phase One:**

File: WRAP_AROUND.M

File: DATA.M
<table>
<thead>
<tr>
<th>File Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SENS_ARRAY_COMB.M</td>
<td>79</td>
</tr>
<tr>
<td>AGREE.M</td>
<td>80</td>
</tr>
<tr>
<td>ALL_AGR.M</td>
<td>81</td>
</tr>
<tr>
<td>TWO_AGR_1_2.M</td>
<td>82</td>
</tr>
<tr>
<td>TWO_AGR_1_3.M</td>
<td>83</td>
</tr>
<tr>
<td>TWO_AGR_2_3.M</td>
<td>84</td>
</tr>
<tr>
<td>NONE_AGR.M</td>
<td>86</td>
</tr>
<tr>
<td>PHASE TWO AND THREE:</td>
<td>87</td>
</tr>
<tr>
<td>DATA_CREATION.M</td>
<td>87</td>
</tr>
<tr>
<td>USER_INTERFACE.M</td>
<td>89</td>
</tr>
<tr>
<td>WO_VAR_DATA_ORG.M</td>
<td>90</td>
</tr>
<tr>
<td>GEN_TEST.M</td>
<td>91</td>
</tr>
<tr>
<td>DEM_NUMER.M</td>
<td>93</td>
</tr>
<tr>
<td>REAL_T_TABLE.M</td>
<td>94</td>
</tr>
<tr>
<td>REAL_T_TABLE_WEIGHTED.M</td>
<td>97</td>
</tr>
</tbody>
</table>
LIST OF TABLES

Table 2.1 Definition of Variables for Target Function and Reconstruction Equation 8

Table 2.2 Variable Assignment for Likelihood Equation 14

Table 2.3 Peak Feature Classifier Data Set\(^6\) 19

Table 2.4 Edge Feature Classifier Data Set\(^6\) 20

Table 2.5 Corner Feature Classifier Data Set\(^6\) 21

Table 3.1. Appropriate Calculated Values for Example 30

Table 6.1 Example Data for Single Target 50

Table 6.2 Example of Classification Agreement and Disagreement 51

Table 6.3 Example of Output of Agreement Algorithm 51

Table 6.4 Permutation Connected to Classification Attempt #1 54

Table 6.5 Masses of Intersection for Classification Attempt # 1 54
LIST OF FIGURES

Figure 2.1 Basic SAR Geometry [2] 5

Figure 2.2 Median Result [6] 10

Figure 2.3 Gaussian Mask and Filter Result [6] 10

Figure 2.4 Outputs at Various Points of the Algorithm [6] 11

Figure 2.5 Line Detection Masks and Corner Extraction Results [6] 11

Figure 2.6 Result of Peak Extraction Algorithm [6] 12

Figure 2.7 Hausdorff Distance Function Example [6] 12

Figure 2.8 Chamfer 3,4 Method Masks [6] 13

Figure 2.9 Confusion Matrix for Peaks Classifier 15

Figure 2.10 Confusion Matrix for Edges Classifier 16

Figure 2.11 Confusion Matrix for Corners Classifier 16

Figure 2.12 Training Set Development [6] 17

Figure 2.13 Total Classification Algorithm [6] 18

Figure 5.1 Rule Selection Dependent on k Value 39

Figure 5.2 Proof that Unified Combination Rule Assumes Yager’s Combination Rule 40

Figure 5.3 Proof that Unified Combination Rule Assumes Dempster’s Combination Rule 40

Figure 5.4 Derivation of Inagaki’s Extra Rule 41

Figure 6.1 Diagram of Data Development 43

Figure 6.2 Diagram of Data Organization 44

Figure 6.3 Diagram of Determination of Classifier Agreement 45

Figure 6.4 Diagram of Dempster’s Combination Rule after Classifier Agreement Determination 46

Figure 6.5 Example of Data Selection 48
Figure 6.6 Permutation Connected to Classification Attempt #4 55
Figure 6.7 Permutation Connected to Classification Attempt #4 with Concatenated Disagreeing Classifier Belief Compliments 55
Figure 6.8 Masses of Intersection for Classification Attempt #4 55
Figure 6.9 Permutation Connected to Classification Attempt #8 56
Figure 6.10 Permutation Connected to Classification Attempt #8 with Concatenated Disagreeing Classifier Belief Compliments 57
Figure 6.11 Masses of Intersection for Classification Attempt #8 57
Figure 6.12 Simple Flow Chart of Dempster-Shafer Theory of Evidence Algorithm 58
Figure 6.13 Rule Selection Dependent on k Value 59
Figure 7.1 Confusion Matrix for the Peaks Feature Classifier 61
Figure 7.2 Confusion Matrix for Un-weighted System 62
Figure 7.3 Confusion Matrix for Weighted System 63
Figure 7.4 Confusion Matrix using Inagaki’s Combination Rule 64
Figure 7.5 Initial Weights by which the Confidence Values are Scaled[6] 65
Figure 7.6 Confusion Matrix as a Result of Initial Weights Shown in Figure 7.5[6] 65
Figure 7.7 Global Weight Scheme[6] 66
Figure 7.8 Confusion Matrix as a Result of Global Weighting[6] 66
Figure 7.9 Weights Determine through Iterative Process[6] 66
Figure 7.10 Confusion Matrix as a Result of Weights Determine through Iterative Process[6] 66
1. INTRODUCTION

The following thesis is part of the research currently being conducted by Professor Sahgri. In the past multiple studies into various ways to perform automatic target recognition have been conducted, all with differing degrees of success. One of those studies also involved the combination of multiple feature based classifiers using classical probability\(^{[6]}\).

The focus of this thesis is the combination of sensors for the purpose of automatic target recognition. Rather than using classical probability, Dempster-Shafer’s Theory of Evidence is used. The first goal of this thesis is to determine whether or not Dempster-Shafer’s Theory of Evidence is applicable to this specific application. The first goal is accomplished through a proof of concept implementation. The second goal is to determine which method is most effective, classical probability or Dempster-Shafer’s Theory of Evidence.
1.1 Document Overview

This document is split into three main parts. The first part is background. The background section addresses the basics of Synthetic Aperture Radar and the development of the data that became the input into Dempster-Shafer’s Theory of Evidence. The second part contains chapters three, four and five and is the theory section. Those three chapters present Dempster-Shafer’s Theory of Evidence, Yager’s Combination Rule and Inagaki’s Unified Combination Rule. The final part of the thesis is specific to the algorithm created. Chapters six, seven and eight detail the implementation, results, and conclusions, respectively.

1.2 Software Platform

MatLab was chosen as the software platform to implement the algorithm created for this thesis. This algorithm is very dependent on matrix manipulation, which MatLab simplifies. In addition, the ability to visually access all variables within the workspace increases the speed of debugging. It is also the programming language that the author of this thesis is the most familiar with. The only reference used for MatLab support was the help application internal to MatLab.
2. BACKGROUND

2.1 Synthetic Aperture Radar

The following section contains a brief overview of Synthetic Aperture Radar (SAR). This section summarizes: the history of SAR, a few SAR basics, SAR geometry, and finally a Generic SAR equation. While the focus of this thesis does not involve SAR, the data utilized is directly related. Therefore it is necessary to have at least a minor understanding of the origins of the data.

2.1.2 Brief Historic Perspective

Original radar systems were designed for military use. Radar was used to track aircraft and ships in low visibility conditions. Original radar systems measured the range to a target and direction via time delay and antenna directivity, respectively. Doppler shifts were first used to measure speed, however Carl Wiley of Goodyear Aerospace (1951) discovered that the Doppler shift could be used to generate fine resolution perpendicular to range\(^{[2]}\). The discovery lead to the realization that two-dimensional
images could be created. The signal processing required creates a very long antenna, effectively. This attribute is the reason that it was called Synthetic Aperture Radar\textsuperscript{[2]}.

The Military released the SAR technology to the civilian community in the 1970’s. The data from SAR looks like random noise. The information that creates the images exists in the phase of the data received. The original SAR systems required Fourier optics to generate the actual images. The received data was recorded on black and white film. A laser beam was collimated and shone through the film. At which point it was processed. Lenses were used to do real time two-dimensional Fourier transforms and diffraction gratings were used to focus resultant images. The original method produced focused images using real time processing, however the Fourier optics required a large optical bench with high quality lenses and a skilled operator to control image quality. Digital SAR processor algorithms were developed in 1978. The range Doppler algorithm was developed by both MacDonald Dettwiler and JPL at the same time, but separately\textsuperscript{[2]}.

2.1.3 SAR Basics

To create the synthetic aperture it is necessary that the radar beam be perpendicular to the direction of travel. SAR transmits phase encoded pulses and records the radar echoes as they reflect off the target/surface. Range is the direction parallel to the direction of the radar beam, time delay is used to measure the distance the response is from the radar and corresponds with the x-axis. The azimuth direction is defined as perpendicular to the range/beam direction, it is parallel to the motion of the sensor itself. The y-axis mapping is determined using the knowledge that the electromagnetic pulses are emitted at a set frequency and the sensor is moving at a continuous rate.
In almost all radar systems pulse compression is required. Without pulse compression the resolution is not fine enough. SAR requires pulse compression. Not only does pulse compression obtain a fine resolution it minimizes peak power and helps maximize the signal-to-noise ratio. Pulse compression is achieved using a match filter.

One of the main problems of SAR is Range Cell Migration. It is a result of the change in range due to the motion of the sensor, which is fundamental to the creation of the synthetic aperture. The processing of the echoes is two dimensional, both the range and azimuth direction. The processing can be separated into independent processes if the received energy does not change significantly over the synthetic aperture. If the range change (range migration) is larger than one sample per cell it must be taken into account in the processing. Range Cell Migration Correction (RCMC) is necessary to maintain the ability to separate the processing[^2].

### 2.1.4 SAR Geometry

Figure 2.1 shows the simple geometric model. Following the figure is a list of all terms seen in the figure and their definition.

![Figure 2.1 Basic SAR Geometry](image)

[^2]: Reference [2]
**Slant Range (Before and After Processing):** The Slant Range is defined as the plane that contains both the relative sensor velocity vector and the slant range vector.

**Beam Footprint:** The Beam Footprint is defined as the area illuminated by the electromagnetic energy transmitted by the radar.

**Plane of Zero Doppler:** The Zero Doppler Plane is defined as a plane that contains the sensor and is perpendicular to the sensors direction of travel. The intersection of this plane with the surface plane is called the Zero Doppler Line.

**Squint Angle:** The Squint Angle is defined as the angle that the Slant Range makes with the zero Doppler plane.

**Ground Range (After Processing to Zero Doppler):** The Ground Range is defined as the projection of the slant range onto the ground.

**Nadir:** Nadir is defined as the point directly below the sensor, normal to the surface.

**Radar Track (Azimuth):** The Radar Track is defined as the route mapped out by nadir as the sensor moves.

### 2.1.5 Brief Generic SAR Equation Overview

The echo of the electromagnetic pulses is often call the SAR signal, the two-dimensional signal is denoted \( s(t, u) \)^{14}. The \( t \) corresponds to the time domain and is related to the time-delay of the echo signal. It is also referred to as the fast-time domain as the SAR signal travels at the speed of light. The \( u \) corresponds to the synthetic aperture domain alternatively referred to as the slow-time domain as the velocity is bounded by the speed of the platform which is significantly less than the speed of light. The SAR signal for the simplified generic system is shown as equation 2.1 and table 2.1 defines all variables.
\[ s(t,u) = \sum_n \sigma_n p \left[ t - \frac{2 \sqrt{x_n^2 + (y_n - u)^2}}{c} \right] \tag{2.1} \]

where

\[ \frac{2 \sqrt{x_n^2 + (y_n - u)^2}}{c} \] \tag{2.2}

is the total delay from radar to \( n \)th target, back to radar. Wavefront reconstruction is used to generate the target function from the SAR signal. Practical reconstruction is done using fast-time matched filtering where the complex conjugate of the Fourier transformed transmitted radar signal is multiplied by the Fourier transformed SAR signal:

\[ F[k_x(\omega, k_u) , k_y(\omega, k_u)] = P^*(\omega)S(\omega, k_u) \tag{2.3} \]

where

\[ k_x(\omega, k_u) = \sqrt{4k^2 - k_u^2}, \]
\[ k_y(\omega, k_u) = k_u \tag{2.4} \]

are two functions called SAR spatial frequency mapping\(^{[14]}\). Which allow for the correct assignment of values in the Cartesian coordinate system.
Table 2.1 Definition of Variables for Target Function and Reconstruction Equation

<table>
<thead>
<tr>
<th>$s(t, u)$</th>
<th>Echo Signal/SAR Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S(\omega, k_u)$</td>
<td>Fourier Transformed SAR Signal</td>
</tr>
<tr>
<td>$f(x,y)$</td>
<td>Target Function</td>
</tr>
<tr>
<td>$F(k_x,k_y)$</td>
<td>Spatial Fourier Transform Target Function</td>
</tr>
<tr>
<td>$p(t)$</td>
<td>Transmitted Radar Signal</td>
</tr>
<tr>
<td>$P(\omega)$</td>
<td>Fourier Transformed Transmitted Radar Signal</td>
</tr>
<tr>
<td>$\sigma_n$</td>
<td>Reflectivity of $n$th target</td>
</tr>
<tr>
<td>$x_n$</td>
<td>Range of $n$th target</td>
</tr>
<tr>
<td>$y_n$</td>
<td>Cross-Range of $n$th target</td>
</tr>
<tr>
<td>$c$</td>
<td>$3 \times 10^8$ m/s (speed of light)</td>
</tr>
<tr>
<td>$k$</td>
<td>Wave Number</td>
</tr>
<tr>
<td>$\omega$</td>
<td>Temporal frequency domain for fast-time $t$</td>
</tr>
<tr>
<td>$k_u$</td>
<td>Spatial frequency domain for azimuthal synthetic aperture</td>
</tr>
</tbody>
</table>

2.2 Data Development

The implementation of the Dempster-Shafer algorithm utilizes the data created from a previous student’s thesis. The results of this thesis will also be compared to that same student’s results. As a consequence this section will be an overview of Chessa Guilas’ thesis, “Hausdorff Probabilistic Feature Analysis in SAR Image Recognition.” Chessa Guilas’ alternative approaches and verification schemes will not be addressed, for all information not included below please review, “Hausdorff Probabilistic Feature
Analysis in SAR Image Recognition.” In this section all figures are from “Hausdorff Probabilistic Feature Analysis in SAR Image Recognition.”[6]

2.2.1 Base Images

The research that was performed in “Hausdorff Probabilistic Feature Analysis in SAR Image Recognition” made use of the Moving and Stationary Target Acquisition and Recognition (MSTAR) publicly available X-band SAR image data set. The data set contains eight known target types at varying depression angles. Images of the targets at depression angles of 15˚ and 17˚ oriented between 270˚ and 359˚ were chosen for analysis in this particular thesis[6]. The eight different target types are titled 2S1, BRDM2, BTR60, D7, SLICY, T62, ZIL_131 (hence forth referred to as ZIL), and ZSU_23_4 (hence forth referred to as ZSU).

2.2.2 Preprocessing

Due to the large amount of Speckle Noise in SAR imagery it is necessary to put the images through preprocessing before feature extraction can be implemented. Median and Gaussian Filtering techniques work well to remove Speckle Noise[6].

2.2.2.1 Median Filtering

The median filter is an order-statistic nonlinear spatial filter[5]. The median filter replaces the center pixel with the median of the intensity values in its given neighborhood. The original value of the center pixel is included in the calculation. The neighborhood is determined by the size of the median filter, in this case it is three by three pixels. Figure 2.2 below is an example of median filtering.
2.2.2.2 Gaussian Filtering

The Gaussian filter is a spatial smoothing filter\textsuperscript{[4]} that is convolved with the image. The Gaussian filter reduces the amount of speckle noise in an image similar to the median filter, however it maintains more detail. Specifically, the edges within the images are less degraded which is fundamental when extracting the edge feature. Figure 2.3 below contains both the Gaussian spatial filter used in Chessa Guillas’ implementation and an example.

![Original Image](Image1)

2.3 Feature Extraction

2.3.1 Edges

The extraction of the edges was done using Canny Edge Detection (for a description of the algorithm see “Hausdorff Probabilistic Feature Analysis in SAR Image Recognition”). The smoothing filter is necessary because a derivative is taken within the Canny Edge Detection Algorithm\textsuperscript{[13]}. A derivative increases the magnitude of the noise with respect to the system. A median filter eliminates too much data so a Gaussian filter is used. The output of the Canny Edge Detection Algorithm is a grayscale image. A
threshold of 128 is then applied to create a binary image. Figure 2.4 displays the output at various points of the algorithm.

![Figure 2.4 Outputs at Various Points of the Algorithm](image)

### 2.3.2 Corners

After applying the median filter a series of four masks are convolved with the image to identify the corners. The four masks are horizontal, vertical, and diagonal (45° and 135°) line identifiers. If a point is identified as an intersection and has a value above the threshold it is stored as a corner. Figure 2.5 contains both the four masks and an example of corner extraction.

![Figure 2.5 Line Detection Masks and Corner Extraction Results](image)

### 2.3.3 Peaks

The maximum in a three by three pixel area is considered a peak if it has an intensity value greater than the threshold of 128. Figure 2.6 is the result of the peaks extraction algorithm.
2.4 Hausdorff Distance

Chessa Guilas used the Hausdorff Distance function to match images. It is defined as the distance between two sets. Letting those two sets be called $A$ and $B$ the following equation is the computation\[^{13}\]:

$$h(A, B) = \max_{a \in A} \min_{b \in B} d(a, b),$$

(2.5)

where $d(a, b)$ is a distance metric. An example is shown in Figure 2.7 below.

2.4.1 Distance Transform

A distance transform takes a binary image and creates a grayscale image where the intensity (0-255) defines the original distance that pixel was from a non-zero value. There are multiple distance metrics that can be used, Chessa Guilas used the Chamfer 3,4 Method. The Chamfer 3,4 Method requires two spatial filters. One is convolved normally, left to right and top to bottom, however the other is convolved for right to left and bottom to top. Both masks are shown in Figure 2.8. The Chamfer 3,4 Method was chosen based on accuracy and efficiency as compared to the City Block and Euclidean distance metrics.
2.4.2 Average Hausdorff Distance and Partial Hausdorff Distance

Regardless of the preprocessing there will still exist some noise in the image. To reduce the effect of the noise on the Hausdorff Distance calculation a slight variant of the calculation is chosen. To have the most robust system Chessa Guilas used the Average Hausdorff Distance with the Partial Hausdorff Distance. The Partial Hausdorff Distance calculation handles the problem of outliers while the Average Hausdorff Distance calculation creates a continuous valued response. The distance calculation is shown in equation 2.6\textsuperscript{[6]}.

\[
H_a(A, B) = \frac{\sum_{i} \{\min\{d_i(a, b)\} < \delta\}}{K},
\]

(2.6)

where A and B are two sets (as previously stated) and

\[
\min\{d_i(a, b)\} < \delta,
\]

(2.7)

is the minimum distance between the two sets less than the threshold value \(\delta\). \(K\) is defined by the total number of distances less than \(\delta\).

2.4.3 Training Set Development

To develop the training set twenty images of each of the eight targets were chosen from the MSTAR dataset. One of the twenty images for each of the targets was designated the template image and was distance transformed. Each of the remaining nineteen images were run through the feature extraction algorithms and superimposed (one at a time) on the template image. The Hausdorff Distance was calculated for each feature of every target. Using those calculated distances a probability function (PF) was
built for each feature of every target and a feature weight assigned. In total twenty-four
PFs were created as seen in Figure 2.12 located at the end of this section. The feature
weight is important to Chessa Guillas’ combination rule and its impact will be seen in the
results chapter of this thesis, however the data that is used for Dempster-Shafer is based
solely on a single feature classifier’s accuracy.

2.4.4 Classification

Ten images of each target are chosen and run through the preprocessing and
feature extraction algorithm. The Hausdorff Distance of the feature extracted images is
compared to the same template image used in the training set development. The
Hausdorff Distance is plotted on the previously created PFs and the corresponding
probability is compared for each of the eight targets. Chessa Guillas uses a maximum
likelihood estimation calculation\(^6\) (Equation 2.8) to classify the targets based on the
probabilities assigned for the PFs and the feature weights as can be seen in Figure 2.13 at
the end of this section.

\[
Likelihood \ of \ Belonging \ to \ Type \ X = (E_{ax}E_x) + (C_{ax}C_x) + (P_{ax}P_x),
\]  

(2.8)

where:

<table>
<thead>
<tr>
<th>Variable Assignment for Likelihood Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( E_{ax} ) = Edge probability function (PF) value for Type X</td>
</tr>
<tr>
<td>( E_x ) = Edge feature weight for Type X</td>
</tr>
<tr>
<td>( C_{ax} ) = Corner probability function (PF) value for Type X</td>
</tr>
<tr>
<td>( C_x ) = Corner feature weight for Type X</td>
</tr>
<tr>
<td>( P_{ax} ) = Peak probability function (PF) value for Type X</td>
</tr>
<tr>
<td>( P_x ) = Peak feature weight for Type X</td>
</tr>
</tbody>
</table>
For the purpose of the data used as the input to Dempster-Shafer the feature weights are set to 100%, 0%, and 0% (peaks, corners, and edges, respectively) where the 100% is rotated through so as to create three data sets with classification based solely on one feature. The data created and used as the input to Dempster-Shafer is shown in tables 2.3, 2.4 and 2.5 located at the end of this section. During the discussion of the implementation of the Dempster-Shafer algorithm it will be necessary to know the three feature classifiers. Peaks has an accuracy of 71.20 percent as shown in the confusion matrix of figure 2.9. Edges has an accuracy of 58.70 percent as shown in the confusion matrix of figure 2.10. Corners has an accuracy of 50 percent as shown in the confusion matrix of figure 2.11.

<table>
<thead>
<tr>
<th></th>
<th>ZSU</th>
<th>ZIL</th>
<th>D7</th>
<th>2S1</th>
<th>SLICY</th>
<th>BDRM2</th>
<th>BTR60</th>
<th>T62</th>
<th>Error of Exclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZSU</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>30%</td>
</tr>
<tr>
<td>ZIL</td>
<td>1</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10%</td>
</tr>
<tr>
<td>D7</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>60%</td>
</tr>
<tr>
<td>2S1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>20%</td>
</tr>
<tr>
<td>SLICY</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>BDRM2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>4</td>
<td>50%</td>
</tr>
<tr>
<td>BTR60</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>60%</td>
</tr>
<tr>
<td>T62</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0%</td>
</tr>
</tbody>
</table>

| Error of Inclusion | 46.15% | 10% | 0% | 0% | 9.09% | 44.44% | 0% | 52.38% | Accuracy | 71.25% |

Figure 2.9 Confusion Matrix for Peaks Classifier
<table>
<thead>
<tr>
<th></th>
<th>ZSU</th>
<th>ZIL</th>
<th>D7</th>
<th>2S1</th>
<th>SLICY</th>
<th>BDRM2</th>
<th>BTR60</th>
<th>T62</th>
<th>Error of Exclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZSU</td>
<td>9</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10%</td>
</tr>
<tr>
<td>ZIL</td>
<td>1</td>
<td>7</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>30%</td>
</tr>
<tr>
<td>D7</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>30%</td>
</tr>
<tr>
<td>2S1</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>60%</td>
</tr>
<tr>
<td>SLICY</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>BDRM2</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>80%</td>
</tr>
<tr>
<td>BTR60</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>60%</td>
</tr>
<tr>
<td>T62</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>60%</td>
</tr>
<tr>
<td>Error of Inclusion</td>
<td>64%</td>
<td>22%</td>
<td>53%</td>
<td>20%</td>
<td>0%</td>
<td>33%</td>
<td>43%</td>
<td>33%</td>
<td>58.70%</td>
</tr>
</tbody>
</table>

**Figure 2.10 Confusion Matrix for Edges Classifier**

<table>
<thead>
<tr>
<th></th>
<th>ZSU</th>
<th>ZIL</th>
<th>D7</th>
<th>2S1</th>
<th>SLICY</th>
<th>BDRM2</th>
<th>BTR60</th>
<th>T62</th>
<th>Error of Exclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZSU</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>80%</td>
</tr>
<tr>
<td>ZIL</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>70%</td>
</tr>
<tr>
<td>D7</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>70%</td>
</tr>
<tr>
<td>2S1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>SLICY</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>BDRM2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>3</td>
<td>60%</td>
</tr>
<tr>
<td>BTR60</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>T62</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>20%</td>
</tr>
<tr>
<td>Error of Inclusion</td>
<td>67%</td>
<td>0%</td>
<td>25%</td>
<td>69%</td>
<td>0%</td>
<td>64%</td>
<td>0%</td>
<td>43%</td>
<td>50%</td>
</tr>
</tbody>
</table>

**Figure 2.11 Confusion Matrix for Corners Classifier**
Figure 2.12 Training Set Development

Selection of 20 Images From Database

Select "Template" Image for Distance Transformation

Filtering: Gaussian and Median

Feature Extraction: Edges, Corners, Peaks

Filtering: Gaussian and Median

Edges Image x 19

Corners Image x 19

Peaks Image x 19

Distance Transform Each feature image is distance transformed (blurred).

Comparison To Distance Transformed Images:
Superimpose, rotate, and obtain average Hausdorff distance from each blurred (distance-transformed) image.

Edges

Corners

Peaks

Average Hausdorff Distance Values x 19 (per feature)

Plot Average Values: For each feature, 19 average values are plotted on a 1-D axis.

Assign Probability Function (PF): A PF is assigned to the plot based on plot density.

Assign Feature Weight: A feature weight is assigned to each plot and is a variable parameter used to optimize recognition accuracy.

Each target type has three plots, one for each feature.
Figure 2.13 Total Classification Algorithm\(^6\)
<table>
<thead>
<tr>
<th>Actual tank num</th>
<th>Actual Image</th>
<th>Prob</th>
<th>Classified As</th>
<th>Image num</th>
<th>Actual tank num</th>
<th>Prob</th>
<th>Classified As</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0]ZSU 14931</td>
<td>ZSU</td>
<td>0.89</td>
<td>ZSU</td>
<td>[4]SLICY 14960</td>
<td>0.9</td>
<td>SLICY</td>
<td></td>
</tr>
<tr>
<td>[0]ZSU 14999</td>
<td>ZSU</td>
<td>0.7</td>
<td>ZSU</td>
<td>[4]SLICY 15227</td>
<td>0.9</td>
<td>SLICY</td>
<td></td>
</tr>
<tr>
<td>[0]ZSU 15065</td>
<td>ZSU</td>
<td>0.7</td>
<td>ZSU</td>
<td>[4]SLICY 19503</td>
<td>0.9</td>
<td>SLICY</td>
<td></td>
</tr>
<tr>
<td>[0]ZSU 15066</td>
<td>T62</td>
<td>0.8</td>
<td>T62</td>
<td>[4]SLICY 19505</td>
<td>0.9</td>
<td>SLICY</td>
<td></td>
</tr>
<tr>
<td>[0]ZSU 15132</td>
<td>T62</td>
<td>0.8</td>
<td>T62</td>
<td>[4]SLICY 19804</td>
<td>0.9</td>
<td>SLICY</td>
<td></td>
</tr>
<tr>
<td>[0]ZSU 15196</td>
<td>ZSU</td>
<td>0.7</td>
<td>ZSU</td>
<td>[4]SLICY 19873</td>
<td>0.9</td>
<td>SLICY</td>
<td></td>
</tr>
<tr>
<td>[0]ZSU 15198</td>
<td>T62</td>
<td>0.8</td>
<td>T62</td>
<td>[4]SLICY 19942</td>
<td>0.9</td>
<td>SLICY</td>
<td></td>
</tr>
<tr>
<td>[0]ZSU 15261</td>
<td>ZSU</td>
<td>0.89</td>
<td>ZSU</td>
<td>[4]SLICY 19943</td>
<td>0.9</td>
<td>SLICY</td>
<td></td>
</tr>
<tr>
<td>[0]ZSU 15265</td>
<td>ZSU</td>
<td>0.89</td>
<td>ZSU</td>
<td>[4]SLICY 20009</td>
<td>0.9</td>
<td>SLICY</td>
<td></td>
</tr>
<tr>
<td>[0]ZSU 15267</td>
<td>ZSU</td>
<td>0.89</td>
<td>ZSU</td>
<td>[4]SLICY 20010</td>
<td>0.9</td>
<td>SLICY</td>
<td></td>
</tr>
<tr>
<td>[1]ZIL 14998</td>
<td>ZIL</td>
<td>0.75</td>
<td>ZIL</td>
<td>[5]BDRM2 14995</td>
<td>0.85</td>
<td>T62</td>
<td></td>
</tr>
<tr>
<td>[1]ZIL 14999</td>
<td>ZIL</td>
<td>0.85</td>
<td>ZIL</td>
<td>[5]BDRM2 15066</td>
<td>0.8</td>
<td>BDRM2</td>
<td></td>
</tr>
<tr>
<td>[1]ZIL 15000</td>
<td>ZIL</td>
<td>0.85</td>
<td>ZIL</td>
<td>[5]BDRM2 15067</td>
<td>0.8</td>
<td>BDRM2</td>
<td></td>
</tr>
<tr>
<td>[1]ZIL 15068</td>
<td>ZIL</td>
<td>0.85</td>
<td>ZIL</td>
<td>[5]BDRM2 15068</td>
<td>0.85</td>
<td>T62</td>
<td></td>
</tr>
<tr>
<td>[1]ZIL 15069</td>
<td>ZSU</td>
<td>0.35</td>
<td>ZSU</td>
<td>[5]BDRM2 15194</td>
<td>0.8</td>
<td>BDRM2</td>
<td></td>
</tr>
<tr>
<td>[1]ZIL 15070</td>
<td>ZIL</td>
<td>0.85</td>
<td>ZIL</td>
<td>[5]BDRM2 15195</td>
<td>0.8</td>
<td>BDRM2</td>
<td></td>
</tr>
<tr>
<td>[1]ZIL 15129</td>
<td>ZIL</td>
<td>0.85</td>
<td>ZIL</td>
<td>[5]BDRM2 15196</td>
<td>0.85</td>
<td>T62</td>
<td></td>
</tr>
<tr>
<td>[1]ZIL 15198</td>
<td>ZIL</td>
<td>0.85</td>
<td>ZIL</td>
<td>[5]BDRM2 15198</td>
<td>0.85</td>
<td>T62</td>
<td></td>
</tr>
<tr>
<td>[1]ZIL 15264</td>
<td>ZIL</td>
<td>0.85</td>
<td>ZIL</td>
<td>[5]BDRM2 15262</td>
<td>0.85</td>
<td>ZIL</td>
<td></td>
</tr>
<tr>
<td>[1]ZIL 15266</td>
<td>ZIL</td>
<td>0.85</td>
<td>ZIL</td>
<td>[5]BDRM2 15262</td>
<td>0.8</td>
<td>BDRM2</td>
<td></td>
</tr>
<tr>
<td>[2]D7 19398</td>
<td>D7</td>
<td>0.58</td>
<td>D7</td>
<td>[6]BTR60 3354</td>
<td>0.8</td>
<td>BDRM2</td>
<td></td>
</tr>
<tr>
<td>[2]D7 19402</td>
<td>T62</td>
<td>0.8</td>
<td>T62</td>
<td>[6]BTR60 3355</td>
<td>0.73</td>
<td>BTR60</td>
<td></td>
</tr>
<tr>
<td>[2]D7 19406</td>
<td>D7</td>
<td>0.65</td>
<td>D7</td>
<td>[6]BTR60 3356</td>
<td>0.8</td>
<td>T62</td>
<td></td>
</tr>
<tr>
<td>[2]D7 19536</td>
<td>ZSU</td>
<td>0.35</td>
<td>ZSU</td>
<td>[6]BTR60 3416</td>
<td>0.8</td>
<td>T62</td>
<td></td>
</tr>
<tr>
<td>[2]D7 19538</td>
<td>SLICY</td>
<td>0.9</td>
<td>SLICY</td>
<td>[6]BTR60 3418</td>
<td>0.8</td>
<td>BDRM2</td>
<td></td>
</tr>
<tr>
<td>[2]D7 19544</td>
<td>ZSU</td>
<td>0.35</td>
<td>ZSU</td>
<td>[6]BTR60 3839</td>
<td>0.72</td>
<td>BTR60</td>
<td></td>
</tr>
<tr>
<td>[2]D7 19821</td>
<td>ZSU</td>
<td>0.35</td>
<td>ZSU</td>
<td>[6]BTR60 3896</td>
<td>0.73</td>
<td>BTR60</td>
<td></td>
</tr>
<tr>
<td>[2]D7 19822</td>
<td>D7</td>
<td>0.65</td>
<td>D7</td>
<td>[6]BTR60 4938</td>
<td>0.73</td>
<td>BTR60</td>
<td></td>
</tr>
<tr>
<td>[2]D7 19960</td>
<td>D7</td>
<td>0.65</td>
<td>D7</td>
<td>[6]BTR60 5006</td>
<td>0.8</td>
<td>BDRM2</td>
<td></td>
</tr>
<tr>
<td>[2]D7 19981</td>
<td>ZSU</td>
<td>0.35</td>
<td>ZSU</td>
<td>[6]BTR60 5637</td>
<td>0.8</td>
<td>BDRM2</td>
<td></td>
</tr>
<tr>
<td>[3]2S1 15001</td>
<td>2S1</td>
<td>0.8</td>
<td>2S1</td>
<td>[7]T62 14995</td>
<td>0.85</td>
<td>T62</td>
<td></td>
</tr>
<tr>
<td>[3]2S1 19444</td>
<td>2S1</td>
<td>0.8</td>
<td>2S1</td>
<td>[7]T62 15065</td>
<td>0.8</td>
<td>T62</td>
<td></td>
</tr>
<tr>
<td>[3]2S1 19781</td>
<td>2S1</td>
<td>0.8</td>
<td>2S1</td>
<td>[7]T62 19446</td>
<td>0.8</td>
<td>T62</td>
<td></td>
</tr>
<tr>
<td>[3]2S1 19845</td>
<td>2S1</td>
<td>0.8</td>
<td>2S1</td>
<td>[7]T62 19448</td>
<td>0.85</td>
<td>T62</td>
<td></td>
</tr>
<tr>
<td>[3]2S1 19850</td>
<td>2S1</td>
<td>0.8</td>
<td>2S1</td>
<td>[7]T62 19845</td>
<td>0.85</td>
<td>T62</td>
<td></td>
</tr>
<tr>
<td>[3]2S1 19913</td>
<td>2S1</td>
<td>0.8</td>
<td>2S1</td>
<td>[7]T62 19846</td>
<td>0.85</td>
<td>T62</td>
<td></td>
</tr>
<tr>
<td>[3]2S1 19914</td>
<td>2S1</td>
<td>0.75</td>
<td>2S1</td>
<td>[7]T62 19913</td>
<td>0.85</td>
<td>T62</td>
<td></td>
</tr>
<tr>
<td>[3]2S1 19918</td>
<td>2S1</td>
<td>0.8</td>
<td>2S1</td>
<td>[7]T62 19914</td>
<td>0.85</td>
<td>T62</td>
<td></td>
</tr>
<tr>
<td>[3]2S1 19983</td>
<td>ZSU</td>
<td>0.35</td>
<td>ZSU</td>
<td>[7]T62 19983</td>
<td>0.8</td>
<td>T62</td>
<td></td>
</tr>
<tr>
<td>Actual tank</td>
<td>Image num</td>
<td>Prob</td>
<td>Classified As</td>
<td>Actual tank</td>
<td>Image num</td>
<td>Prob</td>
<td>Classified As</td>
</tr>
<tr>
<td>------------</td>
<td>-----------</td>
<td>------</td>
<td>---------------</td>
<td>------------</td>
<td>-----------</td>
<td>------</td>
<td>---------------</td>
</tr>
<tr>
<td>[0]ZSU</td>
<td>14931</td>
<td>0.72</td>
<td>ZSU</td>
<td>[4]SLICY</td>
<td>14960</td>
<td>0.95</td>
<td>SLICY</td>
</tr>
<tr>
<td>[0]ZSU</td>
<td>14999</td>
<td>0.68</td>
<td>ZSU</td>
<td>[4]SLICY</td>
<td>15227</td>
<td>0.95</td>
<td>SLICY</td>
</tr>
<tr>
<td>[0]ZSU</td>
<td>15065</td>
<td>0.72</td>
<td>ZSU</td>
<td>[4]SLICY</td>
<td>19503</td>
<td>0.95</td>
<td>SLICY</td>
</tr>
<tr>
<td>[0]ZSU</td>
<td>15066</td>
<td>0.8</td>
<td>D7</td>
<td>[4]SLICY</td>
<td>19505</td>
<td>0.95</td>
<td>SLICY</td>
</tr>
<tr>
<td>[0]ZSU</td>
<td>15132</td>
<td>0.72</td>
<td>ZSU</td>
<td>[4]SLICY</td>
<td>19804</td>
<td>0.9</td>
<td>SLICY</td>
</tr>
<tr>
<td>[0]ZSU</td>
<td>15196</td>
<td>0.72</td>
<td>ZSU</td>
<td>[4]SLICY</td>
<td>19873</td>
<td>0.95</td>
<td>SLICY</td>
</tr>
<tr>
<td>[0]ZSU</td>
<td>15198</td>
<td>0.72</td>
<td>ZSU</td>
<td>[4]SLICY</td>
<td>19942</td>
<td>0.9</td>
<td>SLICY</td>
</tr>
<tr>
<td>[0]ZSU</td>
<td>15261</td>
<td>0.72</td>
<td>ZSU</td>
<td>[4]SLICY</td>
<td>19943</td>
<td>0.95</td>
<td>SLICY</td>
</tr>
<tr>
<td>[0]ZSU</td>
<td>15265</td>
<td>0.72</td>
<td>ZSU</td>
<td>[4]SLICY</td>
<td>20009</td>
<td>0.9</td>
<td>SLICY</td>
</tr>
<tr>
<td>[0]ZSU</td>
<td>15267</td>
<td>0.72</td>
<td>ZSU</td>
<td>[4]SLICY</td>
<td>20010</td>
<td>0.9</td>
<td>SLICY</td>
</tr>
<tr>
<td>[1]ZIL</td>
<td>14998</td>
<td>0.85</td>
<td>ZIL</td>
<td>[5]BDRM2</td>
<td>14995</td>
<td>0.8</td>
<td>BDRM2</td>
</tr>
<tr>
<td>[1]ZIL</td>
<td>14999</td>
<td>0.85</td>
<td>ZIL</td>
<td>[5]BDRM2</td>
<td>15066</td>
<td>0.72</td>
<td>ZSU</td>
</tr>
<tr>
<td>[1]ZIL</td>
<td>15000</td>
<td>0.85</td>
<td>ZIL</td>
<td>[5]BDRM2</td>
<td>15067</td>
<td>0.72</td>
<td>ZSU</td>
</tr>
<tr>
<td>[1]ZIL</td>
<td>15068</td>
<td>0.8</td>
<td>ZIL</td>
<td>[5]BDRM2</td>
<td>15194</td>
<td>0.72</td>
<td>ZSU</td>
</tr>
<tr>
<td>[1]ZIL</td>
<td>15069</td>
<td>0.8</td>
<td>D7</td>
<td>[5]BDRM2</td>
<td>15196</td>
<td>0.8</td>
<td>BDRM2</td>
</tr>
<tr>
<td>[1]ZIL</td>
<td>15070</td>
<td>0.85</td>
<td>ZIL</td>
<td>[5]BDRM2</td>
<td>15198</td>
<td>0.72</td>
<td>ZSU</td>
</tr>
<tr>
<td>[1]ZIL</td>
<td>15129</td>
<td>0.85</td>
<td>ZIL</td>
<td>[5]BDRM2</td>
<td>15261</td>
<td>0.85</td>
<td>ZIL</td>
</tr>
<tr>
<td>[1]ZIL</td>
<td>15264</td>
<td>0.72</td>
<td>ZSU</td>
<td>[5]BDRM2</td>
<td>15262</td>
<td>0.68</td>
<td>BTR60</td>
</tr>
<tr>
<td>[1]ZIL</td>
<td>15266</td>
<td>0.85</td>
<td>ZIL</td>
<td>[5]BDRM2</td>
<td>15262</td>
<td>0.68</td>
<td>BTR60</td>
</tr>
<tr>
<td>[2]D7</td>
<td>19398</td>
<td>0.8</td>
<td>D7</td>
<td>[6]BTR60</td>
<td>3354</td>
<td>0.72</td>
<td>ZSU</td>
</tr>
<tr>
<td>[2]D7</td>
<td>19402</td>
<td>0.8</td>
<td>D7</td>
<td>[6]BTR60</td>
<td>3355</td>
<td>0.49</td>
<td>BTR60</td>
</tr>
<tr>
<td>[2]D7</td>
<td>19406</td>
<td>0.8</td>
<td>D7</td>
<td>[6]BTR60</td>
<td>3356</td>
<td>0.72</td>
<td>ZSU</td>
</tr>
<tr>
<td>[2]D7</td>
<td>19536</td>
<td>0.8</td>
<td>D7</td>
<td>[6]BTR60</td>
<td>3416</td>
<td>0.8</td>
<td>D7</td>
</tr>
<tr>
<td>[2]D7</td>
<td>19538</td>
<td>0.8</td>
<td>2S1</td>
<td>[6]BTR60</td>
<td>3418</td>
<td>0.72</td>
<td>ZSU</td>
</tr>
<tr>
<td>[2]D7</td>
<td>19544</td>
<td>0.8</td>
<td>D7</td>
<td>[6]BTR60</td>
<td>3839</td>
<td>0.72</td>
<td>ZSU</td>
</tr>
<tr>
<td>[2]D7</td>
<td>19821</td>
<td>0.8</td>
<td>D7</td>
<td>[6]BTR60</td>
<td>3896</td>
<td>0.49</td>
<td>BTR60</td>
</tr>
<tr>
<td>[2]D7</td>
<td>19822</td>
<td>0.35</td>
<td>ZIL</td>
<td>[6]BTR60</td>
<td>4938</td>
<td>0.6</td>
<td>BTR60</td>
</tr>
<tr>
<td>[2]D7</td>
<td>19960</td>
<td>0.8</td>
<td>D7</td>
<td>[6]BTR60</td>
<td>5006</td>
<td>0.6</td>
<td>BTR60</td>
</tr>
<tr>
<td>[2]D7</td>
<td>19981</td>
<td>0.72</td>
<td>ZSU</td>
<td>[6]BTR60</td>
<td>5637</td>
<td>0.6</td>
<td>BDRM2</td>
</tr>
<tr>
<td>[3]2S1</td>
<td>15001</td>
<td>0.8</td>
<td>D7</td>
<td>[7]T62</td>
<td>14995</td>
<td>0.72</td>
<td>ZSU</td>
</tr>
<tr>
<td>[3]2S1</td>
<td>19444</td>
<td>0.8</td>
<td>D7</td>
<td>[7]T62</td>
<td>15065</td>
<td>0.72</td>
<td>ZSU</td>
</tr>
<tr>
<td>[3]2S1</td>
<td>19583</td>
<td>0.8</td>
<td>D7</td>
<td>[7]T62</td>
<td>15260</td>
<td>0.6</td>
<td>BTR60</td>
</tr>
<tr>
<td>[3]2S1</td>
<td>19781</td>
<td>0.8</td>
<td>D7</td>
<td>[7]T62</td>
<td>19446</td>
<td>0.72</td>
<td>ZSU</td>
</tr>
<tr>
<td>[3]2S1</td>
<td>19845</td>
<td>0.8</td>
<td>D7</td>
<td>[7]T62</td>
<td>19448</td>
<td>0.8</td>
<td>T62</td>
</tr>
<tr>
<td>[3]2S1</td>
<td>19850</td>
<td>0.8</td>
<td>2S1</td>
<td>[7]T62</td>
<td>19845</td>
<td>0.8</td>
<td>T62</td>
</tr>
<tr>
<td>[3]2S1</td>
<td>19913</td>
<td>0.8</td>
<td>2S1</td>
<td>[7]T62</td>
<td>19846</td>
<td>0.8</td>
<td>T62</td>
</tr>
<tr>
<td>[3]2S1</td>
<td>19914</td>
<td>0.8</td>
<td>2S1</td>
<td>[7]T62</td>
<td>19913</td>
<td>0.72</td>
<td>ZSU</td>
</tr>
<tr>
<td>[3]2S1</td>
<td>19918</td>
<td>0.8</td>
<td>2S1</td>
<td>[7]T62</td>
<td>19914</td>
<td>0.8</td>
<td>T62</td>
</tr>
<tr>
<td>[3]2S1</td>
<td>19983</td>
<td>0.49</td>
<td>BTR60</td>
<td>[7]T62</td>
<td>19983</td>
<td>0.72</td>
<td>ZSU</td>
</tr>
</tbody>
</table>
### Table 2.5 Corner Feature Classifier Data Set[^6]

<table>
<thead>
<tr>
<th>Actual tank</th>
<th>Image num</th>
<th>Prob</th>
<th>Classified As</th>
<th>Actual tank</th>
<th>Image num</th>
<th>Prob</th>
<th>Classified As</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0] ZSU</td>
<td>14931</td>
<td>0.63</td>
<td>ZSU</td>
<td>[4] SLICY</td>
<td>14960</td>
<td>0.9</td>
<td>SLICY</td>
</tr>
<tr>
<td>[0] ZSU</td>
<td>14999</td>
<td>0.68</td>
<td>2S1</td>
<td>[4] SLICY</td>
<td>15227</td>
<td>0.9</td>
<td>SLICY</td>
</tr>
<tr>
<td>[0] ZSU</td>
<td>15065</td>
<td>0.8</td>
<td>2S1</td>
<td>[4] SLICY</td>
<td>19503</td>
<td>0.9</td>
<td>SLICY</td>
</tr>
<tr>
<td>[0] ZSU</td>
<td>15066</td>
<td>0.68</td>
<td>BDRM2</td>
<td>[4] SLICY</td>
<td>19505</td>
<td>0.9</td>
<td>SLICY</td>
</tr>
<tr>
<td>[0] ZSU</td>
<td>15132</td>
<td>0.8</td>
<td>T62</td>
<td>[4] SLICY</td>
<td>19804</td>
<td>0.9</td>
<td>SLICY</td>
</tr>
<tr>
<td>[0] ZSU</td>
<td>15196</td>
<td>0.8</td>
<td>T62</td>
<td>[4] SLICY</td>
<td>19873</td>
<td>0.9</td>
<td>SLICY</td>
</tr>
<tr>
<td>[0] ZSU</td>
<td>15198</td>
<td>0.68</td>
<td>2S1</td>
<td>[4] SLICY</td>
<td>19942</td>
<td>0.9</td>
<td>SLICY</td>
</tr>
<tr>
<td>[0] ZSU</td>
<td>15261</td>
<td>0.8</td>
<td>T62</td>
<td>[4] SLICY</td>
<td>19943</td>
<td>0.9</td>
<td>SLICY</td>
</tr>
<tr>
<td>[0] ZSU</td>
<td>15265</td>
<td>0.68</td>
<td>BDRM2</td>
<td>[4] SLICY</td>
<td>20009</td>
<td>0.9</td>
<td>SLICY</td>
</tr>
<tr>
<td>[0] ZSU</td>
<td>15267</td>
<td>0.63</td>
<td>ZSU</td>
<td>[4] SLICY</td>
<td>20010</td>
<td>0.9</td>
<td>SLICY</td>
</tr>
<tr>
<td>[1] ZIL</td>
<td>14998</td>
<td>0.68</td>
<td>2S1</td>
<td>[5] BDRM2</td>
<td>14995</td>
<td>0.68</td>
<td>BDRM2</td>
</tr>
<tr>
<td>[1] ZIL</td>
<td>14999</td>
<td>0.68</td>
<td>BDRM2</td>
<td>[5] BDRM2</td>
<td>15066</td>
<td>0.65</td>
<td>D7</td>
</tr>
<tr>
<td>[1] ZIL</td>
<td>15000</td>
<td>0.8</td>
<td>2S1</td>
<td>[5] BDRM2</td>
<td>15067</td>
<td>0.8</td>
<td>2S1</td>
</tr>
<tr>
<td>[1] ZIL</td>
<td>15068</td>
<td>0.65</td>
<td>ZIL</td>
<td>[5] BDRM2</td>
<td>15068</td>
<td>0.8</td>
<td>T62</td>
</tr>
<tr>
<td>[1] ZIL</td>
<td>15069</td>
<td>0.68</td>
<td>2S1</td>
<td>[5] BDRM2</td>
<td>15194</td>
<td>0.68</td>
<td>BDRM2</td>
</tr>
<tr>
<td>[1] ZIL</td>
<td>15070</td>
<td>0.65</td>
<td>ZIL</td>
<td>[5] BDRM2</td>
<td>15195</td>
<td>0.68</td>
<td>BDRM2</td>
</tr>
<tr>
<td>[1] ZIL</td>
<td>15129</td>
<td>0.8</td>
<td>2S1</td>
<td>[5] BDRM2</td>
<td>15196</td>
<td>0.8</td>
<td>T62</td>
</tr>
<tr>
<td>[1] ZIL</td>
<td>15198</td>
<td>0.68</td>
<td>2S1</td>
<td>[5] BDRM2</td>
<td>15198</td>
<td>0.8</td>
<td>2S1</td>
</tr>
<tr>
<td>[1] ZIL</td>
<td>15264</td>
<td>0.8</td>
<td>2S1</td>
<td>[5] BDRM2</td>
<td>15261</td>
<td>0.8</td>
<td>T62</td>
</tr>
<tr>
<td>[1] ZIL</td>
<td>15266</td>
<td>0.65</td>
<td>ZIL</td>
<td>[5] BDRM2</td>
<td>15262</td>
<td>0.68</td>
<td>BDRM2</td>
</tr>
<tr>
<td>[2] D7</td>
<td>19398</td>
<td>0.65</td>
<td>D7</td>
<td>[6] BTR60</td>
<td>3354</td>
<td>0.68</td>
<td>BDRM2</td>
</tr>
<tr>
<td>[2] D7</td>
<td>19402</td>
<td>0.68</td>
<td>2S1</td>
<td>[6] BTR60</td>
<td>3355</td>
<td>0.63</td>
<td>ZSU</td>
</tr>
<tr>
<td>[2] D7</td>
<td>19406</td>
<td>0.63</td>
<td>ZSU</td>
<td>[6] BTR60</td>
<td>3356</td>
<td>0.8</td>
<td>2S1</td>
</tr>
<tr>
<td>[2] D7</td>
<td>19536</td>
<td>0.8</td>
<td>2S1</td>
<td>[6] BTR60</td>
<td>3416</td>
<td>0.8</td>
<td>2S1</td>
</tr>
<tr>
<td>[2] D7</td>
<td>19538</td>
<td>0.65</td>
<td>D7</td>
<td>[6] BTR60</td>
<td>3418</td>
<td>0.68</td>
<td>BDRM2</td>
</tr>
<tr>
<td>[2] D7</td>
<td>19544</td>
<td>0.68</td>
<td>2S1</td>
<td>[6] BTR60</td>
<td>3839</td>
<td>0.8</td>
<td>2S1</td>
</tr>
<tr>
<td>[2] D7</td>
<td>19821</td>
<td>0.65</td>
<td>D7</td>
<td>[6] BTR60</td>
<td>3896</td>
<td>0.6</td>
<td>ZSU</td>
</tr>
<tr>
<td>[2] D7</td>
<td>19822</td>
<td>0.68</td>
<td>2S1</td>
<td>[6] BTR60</td>
<td>4938</td>
<td>0.68</td>
<td>BDRM2</td>
</tr>
<tr>
<td>[2] D7</td>
<td>19960</td>
<td>0.8</td>
<td>2S1</td>
<td>[6] BTR60</td>
<td>5006</td>
<td>0.63</td>
<td>ZSU</td>
</tr>
<tr>
<td>[2] D7</td>
<td>19981</td>
<td>0.68</td>
<td>2S1</td>
<td>[6] BTR60</td>
<td>5637</td>
<td>0.68</td>
<td>BDRM2</td>
</tr>
<tr>
<td>[3] 2S1</td>
<td>15001</td>
<td>0.8</td>
<td>2S1</td>
<td>[7] T62</td>
<td>14995</td>
<td>0.8</td>
<td>T62</td>
</tr>
<tr>
<td>[3] 2S1</td>
<td>19444</td>
<td>0.8</td>
<td>2S1</td>
<td>[7] T62</td>
<td>15065</td>
<td>0.8</td>
<td>T62</td>
</tr>
<tr>
<td>[3] 2S1</td>
<td>19583</td>
<td>0.8</td>
<td>2S1</td>
<td>[7] T62</td>
<td>15260</td>
<td>0.8</td>
<td>2S1</td>
</tr>
<tr>
<td>[3] 2S1</td>
<td>19781</td>
<td>0.8</td>
<td>2S1</td>
<td>[7] T62</td>
<td>19446</td>
<td>0.8</td>
<td>T62</td>
</tr>
<tr>
<td>[3] 2S1</td>
<td>19845</td>
<td>0.8</td>
<td>2S1</td>
<td>[7] T62</td>
<td>19448</td>
<td>0.8</td>
<td>T62</td>
</tr>
<tr>
<td>[3] 2S1</td>
<td>19850</td>
<td>0.8</td>
<td>2S1</td>
<td>[7] T62</td>
<td>19845</td>
<td>0.8</td>
<td>T62</td>
</tr>
<tr>
<td>[3] 2S1</td>
<td>19913</td>
<td>0.8</td>
<td>2S1</td>
<td>[7] T62</td>
<td>19846</td>
<td>0.8</td>
<td>T62</td>
</tr>
<tr>
<td>[3] 2S1</td>
<td>19914</td>
<td>0.8</td>
<td>2S1</td>
<td>[7] T62</td>
<td>19913</td>
<td>0.8</td>
<td>T62</td>
</tr>
<tr>
<td>[3] 2S1</td>
<td>19918</td>
<td>0.8</td>
<td>2S1</td>
<td>[7] T62</td>
<td>19914</td>
<td>0.68</td>
<td>2S1</td>
</tr>
<tr>
<td>[3] 2S1</td>
<td>19983</td>
<td>0.8</td>
<td>2S1</td>
<td>[7] T62</td>
<td>19983</td>
<td>0.8</td>
<td>T62</td>
</tr>
</tbody>
</table>
3. DEMPSTER-SHAFER’S THEORY OF EVIDENCE

3.1 Introduction to Dempster-Shafer’s Theory of Evidence

Glen Shafer attempted to model ‘belief’ using the mathematic work of A. P. Dempster (1968). In the foreword to “A Mathematical Theory of Evidence” A. P. Dempster states that Glen Shafer has reengineered his original theory around the idea of combining support functions and their corresponding weights of evidence. As a consequence Dempster’s mathematical theory became applicable to more than just random sampling. What is now known as Dempster-Shafer’s Theory of Evidence is both a theory of evidence and a theory of probable reasoning\(^{11}\).

*Theory of Evidence*: A theory that works with weights of evidence and with numerical support based on evidence.
Theory of Probable Reasoning: A theory with a focal point on the fundamental operation of probable reasoning: the combination of evidence.

To understand Dempster-Shafer’s Theory of Evidence it is beneficial to detail a comparison to Bayesian probability. Dempster-Shafer’s Theory of Evidence can be considered a generalization of classical Bayesian probability\(^{[12]}\). If there is adequate evidence such that probabilities can be assigned in the traditional sense Dempster-Shafer will simplify to Bayesian probability\(^{[10]}\). Dempster-Shafer uses ‘belief’ rather than probability, inherently increasing the flexibility. The measure of ‘belief’ allows vague states to exist whereas Bayesian probability grants weight to both a single event and its compliment\(^{[9]}\).

*Example 1:* A sensor focused on a given object identifies it as an airplane. The sensor reports an accuracy of ninety-five percent. According to Bayesian probability the sensor is simultaneously reporting that there exists a five percent chance that the object is *not* an airplane. When Dempster-Shafer is used the accuracy of ninety-five percent is a belief. Unlike Bayesian probability the remaining five percent can be assigned to any other state, be that another object type (tank, dog, person, etc.) or an unknown state.

Dempster-Shafer does not assign weight to states ignorant of evidence. Furthermore, evidence is not restricted in application to a single event unlike Bayesian probability. In the case that some new evidence narrows the hypothesis, Bayesian probability invokes the Principle of Insufficient Reason assigning equal priors to the
remaining elements of the hypothesis set. Shafer claimed that the Principle of Insufficient Reason creates a false degree of knowledge\textsuperscript{[11]}. Not to say that Bayesian probability cannot return a result that is ‘uncertain.’ Given a case where the end result shows equal likelihood between all elements uncertainty can be concluded. Dempster-Shafer, however, explicitly measures uncertainty. Calculation requires an unknown state to exist. One of the strengths of Dempster-Shafer theory is the ability to model the narrowing of a hypothesis set with the accumulation of evidence\textsuperscript{[8]}.

\textit{Example 2:} Given a hypothesis set consisting of the tank, airplane and unknown state a sensor conveys the evidence that the object in question is fast with an accuracy of sixty percent. With Bayesian probability this evidence would imply that there is a sixty percent chance that the object is a plane. The remaining forty percent would be equally placed on the other two states, regardless that there existed no evidence to verify that assumption. Using Dempster-Shafer that forty percent would be assigned to the unknown state as there is no evidence to support any other conclusion in particular.

\textbf{3.2 Basic Probability Assignments}

There are three functions related to Dempster-Shafer’s Theory of evidence, the Basic Probability Assignment function (bpa), the Belief function (Bel), and the Plausibility function (Pl). In this section the bpa is thoroughly detailed. The bpa, represented by \( m \), is often thought of as Bayesian probability. While the bpa can be equivalent to Bayesian probability it does not have to be. As such, it is more correct to say that the bpa is a generalized Bayesian probability. The bpa defines a mapping of the
power set, \(2^X\), to the values zero to one (Equation 3.1). The bpa of the null set is zero (Equation 3.2). This is a reflection that no belief should be assigned to the null set. The sum of all basic probability assignments (bpa’s) of all the subsets of the power set is 1 (Equation 3.3), which embodies the concept that total belief has to be one\(^{10}[15]\).

\[
\begin{align*}
m : P(X) &\to [0,1] \\
m(\emptyset) &= 0 \\
\sum_{A \in P(X)} m(A) &= 1
\end{align*}
\]  \hspace{1cm} (3.1) \hspace{1cm} \hspace{1cm} (3.2) \hspace{1cm} \hspace{1cm} (3.3)

It is important to note that \(m(A)\) measures the belief that is committed exactly to \(A\) and makes no specific claims to any particular subsets of \(A\). Information about specific subsets of \(A\) would be represented by another bpa, \(m(B)\) such that \(B \subset A\).

Once again, given the definition above, the difference in the way Bayesian probability and Dempster-Shafer Theory take care of ignorance is reinforced. Without any evidence \(m(X)\) is given a value of 1 and \(m(A)\) is set to zero for all \(A\) not equal to \(X\). Simply stated a \(m(X)\) of one shows that there is no doubt that the answer exists within the set, but it lends weight to no other conclusion. The bpa in this particular case was entitled the Vacuous bpa by Glen Shafer\(^8\). A system utilizing Bayesian probability would assign equal probabilities to each set. This is an assumption of information without evidence.

\textit{Example 3.} Suppose evidence is presented that supports the concept that the object of interest is a tank, when it could have also been identified as a plane or human. The degree of confidence regarding the evidence is 0.7.

The bpa, \(m(tank)\), is assigned 0.7. The remaining ‘belief’ is assigned to \(m(X) = \{tank, plane, human\} = 0.3\). Bayesian probability would have assigned the remaining 0.3 to \(m(not \ tank)\). However, no evidence had been
presented justifying that claim, where as Dempster-Shafer simply assigns it to the universal set expressing the uncertainty in the system.

3.3 Belief Functions

In the case of a hypothesis set with multiple propositions it is likely that evidence can be found to support one of the propositions. The given evidence may also lend support to another proposition in the case that that proposition contains the proposition in question. A basic probability assignment solves for the mass or belief assigned exactly to one of the propositions as was discussed in the previous section. In this section the focus is belief functions. A belief function looks at the total belief in the proposition including those that are implied.

To find the belief function, \( \text{Bel}(A) \), the measures \( m(B) \) have to be added to the bpa \( m(A) \) where \( m(B) \) is defined for all proper subsets \( B \) of \( A \), which is shown by equation 5\(^{[11]} \). The belief function like the basic probability assignment defines a mapping of the power set, \( 2^X \), to the values zero to one (Equation 3.4).

\[
\text{Bel} : P(X) \rightarrow [0,1]
\]

\[
\text{Bel}(A) = \sum_{B \subseteq A} m(B)
\]

In order to be classified as a belief function \( \text{Bel}(A) \) must adhere to multiple premises.

\[
\text{Bel}(\emptyset) = 0
\]

\[
\text{Bel}(X) = 1
\]

\[
\text{Bel}(A_1, \ldots, A_n) \geq \sum_{I = \{i, \ldots, n\}} (-1)^{|I|+1} \text{Bel}\left(\bigcap_{i \in I} A_i\right)
\]

In the case that the belief functions are the only available information it is possible to recover the basic probability assignments using equation 3.9.
Belief functions exist both in Dempster-Shafer and classical probability. Belief functions can be used to show the difference between Dempster-Shafer’s Theory of evidence and Bayesian probability. In order for the belief function to be Bayesian the following conditions must be met.

\[
Bel(\emptyset) = 0 \quad (3.10)
\]

\[
Bel(X) = 1 \quad (3.11)
\]

The first two conditions are the same as that required for Dempster-Shafer belief functions. It is in the third condition that the difference occurs. That condition is called Bayes’ Rule of Additivity\[^{[11]}\].

\[
Bel(A \cup B) = Bel(A) + Bel(B) \quad (3.12)
\]

Whenever \( A, B \subseteq \mathcal{P}(X) \) and \( A \cap B = \emptyset \)

While it is stated that this displays the difference between Dempster-Shafer and Bayesian probability let it be known that Bayesian belief functions are still Dempster-Shafer belief functions as they obey all the necessary premises expressed in equations 3.6 to 3.8.

However not all Dempster-Shafer belief functions meet the conditions required to be classified as Bayesian belief functions. Bayesian belief functions are defined as a proper subclass of Dempster-Shafer belief functions.

### 3.4 Belief Intervals

In “A Mathematical Theory of Evidence” Glen Shafer claims that the belief in a specific proposition of the hypothesis set is not enough to fully detail the level of certainty the evidence has allowed it to obtain\[^{[11]}\]. It is necessary to know the level of
doubt assigned to the belief. Given the belief function, \( Bel(A) \) the degree of doubt is denoted \( Dou(A) \).

\[
Dou(A) = Bel(\overline{A}) \tag{3.13}
\]

What has become known as the plausibility \( (Pl(A) \) of the proposition was earlier called the upper probability and was denoted \( P^*(A) \). This value represents the plausibility of the proposition hence the change of name.

\[
Pl(A) = P^* (A) \tag{3.14}
\]
\[
Pl(A) = 1 − Dou(A) \tag{3.15}
\]
\[
Pl(A) = 1 − Bel(\overline{A}) \tag{3.16}
\]

\[
Pl(A) = 1 − Bel(\overline{A}) = \sum_{B \in A} m(B) − \sum_{B \in \emptyset} m(B) = \sum_{B \cap A \neq \emptyset} m(B) \tag{3.17}
\]

Depending on the application the final output of Dempster-Shafer could be an interval. The interval contains both the belief and the plausibility of some proposition \( A \).

\[
\begin{bmatrix}
Bel(A) & 1 − Bel(\overline{A})
\end{bmatrix} \tag{3.18}
\]
\[
\begin{bmatrix}
Bel(A) & Pl(A)
\end{bmatrix} \tag{3.19}
\]

### 3.5 Combination

For a given frame of discernment it is possible for multiple sources to provide evidence. In order for them to be combined using Dempster-Shafer it is necessary that all sources be independent. Dempster-Shafer combines belief functions by utilizing the basic probability assignments. The numerator of the combination rule is the summation of all intersections between the propositions being compared whereas the denominator is one minus the summation of all null sets.
The denominator of the combination rule is called the renormalizing constant \( K \). It assigns all conflict to the null set thereby ignoring all conflict in the system. However, the denominator still contains information that measures the quantity of conflict.

\[
K = \left( 1 - \sum_{B \cap C = \emptyset} m_1(B)m_2(C) \right)^{-1} \tag{3.21}
\]

The weight of the conflict between the beliefs in regard to the propositions is denoted \( Con(Bel_1,Bel_2) \) and is determined from the denominator.

\[
Con(Bel_1, Bel_2) = \log(K)
\]

\[
= \log \left( \frac{1}{1 - \sum_{B \cap C = \emptyset} m_1(B)m_2(C)} \right) \tag{3.22}
\]

\( Con(Bel_1,Bel_2) \) can have a value ranging from zero to infinity\(^{[11]} \). A complete lack of conflict in the system is represented by the weight of conflict assuming a value of zero. In the case that there exists nothing but contradictions within the system a value of infinity is assigned to \( Con(Bel_1,Bel_2) \). This relation is not bound to only two elements but can be expanded to match the system in question.

### 3.6 Dangers of Dempster-Shafer

Due to the renormalization constant the Dempster-Shafer combination Rule can lead to counterintuitive results. The counterintuitive outcome is a direct result of ignoring conflict\(^{[10]} \). While this only occurs in cases of extreme conflict it does not allow Dempster-Shafer to be used carelessly. All results have to be checked for the potential of
such an occurrence. The appearance of this potential pitfall has spurred modification to
Dempster-Shafer to create a more robust combination rule. Both Inagaki’s Unified
Combination Rule and Yager’s Modified Dempster-Shafer Combination Rule have been
created and will be further explored in following chapters.

Example 4: Suppose that a patient is diagnosed by two doctors regarding
the patient’s symptoms. The first doctor claims that the patient has either
meningitis with a probability of 0.99 or a brain tumor, with a probability of
0.01. The second doctor states the patient actually suffers from a
concussion with a probability of 0.99 but concedes to the possibility of a
brain tumor with a probability of 0.01.

$$ m(X) = \{ \text{meningitis, brain tumor, concussion} \} $$  \hspace{1cm} (3.23)

$$ m_1(\text{meningitis}) = m_1(m) $$
$$ m_1(\text{brain tumor}) = m_1(b) $$  \hspace{1cm} (3.24)
$$ m_2(\text{concussion}) = m_2(c) $$
$$ m_2(\text{brain tumor}) = m_2(b) $$

Table 3.1. Appropriate Calculated Values for Example

<table>
<thead>
<tr>
<th></th>
<th>$m_2(c)=0.99$</th>
<th>$m_2(b)=0.01$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_1(m)$= 0.99</td>
<td>0.9801</td>
<td>0.0099</td>
</tr>
<tr>
<td>$m_1(b)$= 0.01</td>
<td>0.0099</td>
<td>0.0001</td>
</tr>
</tbody>
</table>

$$ K = 1 - (0.9801 + 0.0099 + 0.9999) = 0.0001 $$ \hspace{1cm} (3.25)

The proposition of a brain tumor is the only intersection. Utilizing equation 3.20
Dempster-Shafer’s Combination rule results in a belief function corresponding to the
brain tumor equal to one. Dempster-Shafer’s Combination Rule is stating that there exists
no doubt that the patient has a brain tumor. Based on the evidence this is clearly wrong.
4. YAGAR’S MODIFIED DEMPSTER-SHAFER

4.1 Introduction

As mentioned in section six of chapter three, there exists a chance of counterintuitive results when using the unmodified Dempster’s Combination Rule. Multiple modifications to Dempster’s Combination Rule have been explored; one such modification was created by Ronald R. Yager. Yager focused on two different weaknesses of the combination rule. The first problem was the addition of new data. With Dempster’s Combination Rule it is necessary to completely recalculate the basic probability assignments that are affected by the addition of new information into the system. This is a result of the non-associative properties of the basic probability assignment. As a result Yager developed the application of quasi-associative operations in the combination of evidence (explained in the following section). Further Yager described what is called Yager’s Modified Dempster-Shafer as an alternative to
Dempster’s Combination Rule to avoid the second weakness, the occurrence of counterintuitive results.

4.2 Quasi-Associative Operators

When fusing data the ability to continuously update the output in response to new evidence without complete recalculation of parameters is desired. This is possible given that the operator involved is associative\footnote{17}. For the purpose of this chapter let $\diamondsuit$ be defined as a binary operator and $X$ a set of elements. $\diamondsuit$ is considered an associative operator if

$$a \diamondsuit (b \diamondsuit c) = (a \diamondsuit b) \diamondsuit c; \text{ for any } a, b, c \in X.$$  \hfill (4.1)

The specific associative property that is key to the combination of evidence is shown below. First let $x_1, x_2, \ldots, x_n, x_{n+1}$ be the elements of $X$.

$$x = x_1 \diamondsuit x_2 \diamondsuit \cdots \diamondsuit x_n$$
$$d = x_1 \diamondsuit x_2 \diamondsuit \cdots \diamondsuit x_n \diamondsuit x_{n+1}.$$  \hfill (4.2)

$$\therefore d = x \diamondsuit x_{n+1}$$

This property allows for the continuous update that is desired in data fusion systems.

Now that the pursuit of an operator that is associative has been justified, the obvious problem is that in most real world combination the operator is not associative. Dempster’s Combination Rule is no exception to that claim. Yager stated that while this is true there are quite a few of these non-associative operators that can be expressed as quasi-associative operators. Essentially, a quasi-associative operator responds to additional information in a similar way as an associative operator. In order to define a quasi-associative operator let it be denoted by $\psi$. Yager uses the following notation to define a quasi-associative operator\footnote{17}:
where $T$ is called a normalization mapping.

The easiest way to interpret the above equation is to simply say there are a set of non-associative operators that can be broken down into sub associative operators. One of the most recognizable operators that fall within this category is the arithmetic average\textsuperscript{[17]} denoted $y$ below. It is non-associative in that averaging a new data point with a preexisting average leads to an insignificant result.

$$\psi = \left( \frac{1}{n} \sum_{i=1}^{n} x_i, n \right) + x/(n+1) \quad (4.4)$$

Where $x_i$ is the data point and $n$ is the number of data points in the system. However if you add the new data point to the sum of the original data points

$$x_1 \triangle x_2 \ldots \triangle x_n \triangle x_{n+1} = x_1 + x_2 + \ldots + x_n + x_{n+1} \quad (4.5)$$

and then divide by the total number of points, where the division would be called the normalization mapping, the operation would then be quasi-associative. The arithmetic average is denoted as:

$$y = T \left( \sum_{i=1}^{n} x_i, n \right) \quad (4.6)$$

Replacing the $T$

$$y = \left( \frac{1}{n} \sum_{i=1}^{n} x_i, n \right) \quad (4.7)$$

Though the above example is simple it concisely shows the advantage of such an operator in comparison to the non-associative alternative. The next logical step is to apply the above framework to Dempster’s Combination Rule, which Yager does.
4.2.1 Quasi-Associative Framework Applied to Dempster’s Combination Rule

Before continuing it is necessary to recall Dempster’s Combination Rule.

\[
m_{12}(A) = \frac{\sum_{B \cap C = A} m_1(B)m_2(C)}{1 - \sum_{B \cap C = \emptyset} m_1(B)m_2(C)}
\]  

(4.8)

This operator, like the arithmetic mean, can be created from sub associative operators.

Yager defines the sub associative operator:

\[
q(A) = \sum_{B_i \cap C_j = A} \left[ m_1(B_i) \times m_2(C_j) \right]_p
\]

where \( a = q(\emptyset) \)

Yager’s goal is to modify the combination rule enough to guarantee that counterintuituitive results will not occur. Instead of the basic probability assignment that Dempster used, Yager implements the ground probability assignment. The ground probability assignment was already introduced in the previous section, however, informally. The ground probability assignment is shown in equation

The above modification to Dempster’s Combination Rule only corrects one of the two identified weaknesses. This allows for continuous update to the system without complete recalculation. In the next section the second weakness will be addressed.

4.3 Yager’s Modified Dempster Shafer

In the last section the modification to create a quasi-associative operator finished just short of the total modification. Yager’s goal is to modify the combination rule enough to guarantee that counterintuituitive results will not occur. Instead of the basic probability assignment that Dempster used, Yager implements the ground probability assignment. The ground probability assignment was already introduced in the previous section, however, informally. The ground probability assignment is shown in equation
4.9. The ground probability assignment is constrained similarly to the basic probability assignment.

\[
q : P(X) \rightarrow [0,1]
\]  \hfill (4.12)

Yager’s combination rule is formally defined in equations 4.13 to 4.15.

\[
m(\emptyset) = 0 \hfill (4.13)
\]

\[
m(X) = q(X) + q(\emptyset) \hfill (4.14)
\]

\[
m(A) = q(A) \hfill (4.15)
\]

Where \( X \) is the universal set and \( q(\emptyset) \) is the degree of conflict of the elements that are being combined. The degree of conflict \( q(\emptyset) \) is calculated in the exact same way as the null set for Dempster’s Combination Rule\textsuperscript{[10][17]}.

\[
q(\emptyset) = \sum_{B \cap C = \emptyset} m_1(B)m_2(C) \hfill (4.16)
\]

The first thing to be noted is that Yager makes uses of the quasi-associative property in his final modification. The second point to note is the true difference between Dempster’s and Yager’s combination rules. It is in their respective treatment of conflict. Dempster multiplies what is essentially Yager’s ground probability assignment by the renormalizing constant \((K)\) effectively removing the conflict from the system.

\[
Recall, \ K = \left(1 - \sum_{B \cap C = \emptyset} m_1(B)m_2(C)\right)^{-1} \hfill (4.17)
\]

Whereas Yager adds the conflict to the universal set which can be thought of as the degree of ignorance within the system. In other words, rather than ignoring the conflict Yager keeps that information by \textit{increasing} the degree of ignorance found in the system. Regardless, the two different combination rules will lead to the same results given there exists no conflicting evidence. Conversely, when there is only conflicting data
Dempster’s Rule cannot be used but Yager’s Rule creates a vacuous belief structure. A vacuous belief structure is defined as a system with 1 assigned to the universal set and 0 assigned to the ground probability assignment. It is a system in which there exists only ignorance.

Yager accomplishes his goal as can be seen by applying the modified combination rule to the example presented in section 6 of chapter 3. The overall belief in the hypothesis of a brain tumor is .0001 and the degree of ignorance in the system is .9999\[16\]. This is the expected result.
5. INAGAKI’S UNIFIED COMBINATION RULE

5.1 Introduction

The method presented in this chapter was developed by Toshiyuki Inagaki. From the last two chapters it is apparent that the outputs of different combination rules are not guaranteed to be the same. Depending on the application the differing outputs can have vastly different consequences. Inagaki proved the truth of this in “Interdependence between Safety-Control Policy and Multiple-Sensor Schemes via Dempster-Shafer Theory.”[7] The systems used for analysis were complex plants (nuclear reactors, spacecraft, etc.) and the decisions were between continued operation and safe shut-down. Inagaki created the Unified Combination Rule, in order to have one combination rule that could be used in all situations[7].
5.2 Unified Combination Rule

Inagaki stated that all valid combination rules could be represented by equation 5.1\textsuperscript{[7][10]}. The function represented by \( f(C) \) is in place to scale the conflict as desired and is defined in equation 5.2.

\[
m(C) = q(C) + f(C)q(\emptyset)
\]

where \( C \neq \emptyset \)

\[
\sum_{C \subseteq X, C \neq \emptyset} f(C) = 1
\]

\[
f(C) \geq 0
\]

(5.2)

The Unified Combination Rule utilizes aspects of both Dempster’s Combination Rule and Yager’s Combination Rule, specifically, the final mapping to a basic probability assignment (\( m \)) and the use of ground probability assignments (\( q \)). Inagaki unifies a specific class of combination rules bound by the ratio expressed in equation 5.3.

\[
\frac{m(C)}{m(D)} = \frac{q(C)}{q(D)}
\]

(5.3)

Equation 5.3 expresses that the only knowledge of the system available is the ground probability assignment and can be satisfied by either Dempster’s Combination Rule or Yager’s Combination Rule\textsuperscript{[7][10]}. Any knowledge of the credibility of the evidence/sources would cause equation 5.3 to be false. The ratio can be written as shown in equation 5.4.

\[
\frac{[q(C) + f(C)q(\emptyset)]}{q(C)} = \frac{[q(D) + f(D)q(\emptyset)]}{q(D)}
\]

(5.4)

Equation 5.4 leads to the property expressed in equation 5.5, where \( k \) is the conflict of the system.
The formal form of the Unified Combination Rule is shown in equations 5.6 and 5.7.

\[
\frac{f(C)}{q(C)} = k \quad \text{for any } C \neq X, \emptyset
\]  

\[
m(C) = [1 + k \, q(\emptyset)]q(C), \, C \neq X, \emptyset
\] 

\[
m(X) = [1 + k \, q(\emptyset)]q(X) + [1 + k \, q(\emptyset) - k]q(\emptyset)
\] 

The value that \( k \) is given determines which of the combination rules the Unified Combination Rule becomes. The value of \( k \) is bounded as shown in equation 5.8.

\[
0 \leq k \leq \frac{1}{[1 - q(\emptyset) - q(X)]}
\] 

There are three different combination rules that the Unified Combination Rule can take on are, Yager’s Combination Rule, Dempster’s Combination Rule, and Inagaki’s Extra Rule. The value of \( k \) necessary to become any of those rules is shown in figure 5.1 below.

It is important to note that \( k \) is not restricted to assuming a value such that one of the known combination rules is implemented. Inagaki claims that the correct value of \( k \) for any given system is a topic for research\[7][10].

Here is a table showing the values of \( k \) for each rule:

<table>
<thead>
<tr>
<th>Yager’s Rule</th>
<th>Dempster-Shafer’s Rule</th>
<th>Inagaki’s Extra Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>( k )</td>
</tr>
<tr>
<td></td>
<td>( \frac{1}{1 - q(\emptyset)} )</td>
<td>( \frac{1}{1 - q(\emptyset) - q(X)} )</td>
</tr>
</tbody>
</table>

**Figure 5.1 Rule Selection Dependent on \( k \) Value**

Figures 5.2 and 5.3 below are proofs that the values of \( k \) shown in figure 5.1 above do in fact lead to their corresponding combination rule.
with $k = 0$:
\[
m(C) = \left\{1 + 0[q(\emptyset)]\right\}q(C) \\
= q(C)
\]
\[
m(X) = \left\{1 + 0[q(\emptyset)]\right\}q(X) + \left\{1 + 0[q(\emptyset)] - 0\right\}q(\emptyset) \\
= q(X) + q(\emptyset)
\]

Figure 5.2 Proof that Unified Combination Rule Assumes Yager’s Combination Rule

with $k = \frac{1}{1 - q(\emptyset)}$:
\[
m(C) = \left\{1 + \left[\frac{1}{1 - q(\emptyset)}\right]q(\emptyset)\right\}q(C) \\
= \frac{q(C)}{1 - q(\emptyset)}
\]
\[
m(X) = \left\{1 + \left[\frac{1}{1 - q(\emptyset)}\right]q(\emptyset)\right\}q(X) + \left\{1 + \left[\frac{1}{1 - q(\emptyset)}\right]q(\emptyset) - \frac{1}{1 - q(\emptyset)}\right\}q(\emptyset) \\
= \frac{q(X)}{1 - q(\emptyset)}
\]

Figure 5.3 Proof that Unified Combination Rule Assumes Dempster’s Combination Rule

5.3 Inagaki’s Extra Rule

Inagaki’s Extra Rule is implemented when $k$ assumes the value equal to its upper boundary. Figure 5.4 contains a short derivation of the rule for both the values of $m(C)$ and $m(X)$. To derive the equations the upper boundary of $k$ is plugged into equations 5.6 and 5.7. Comparing the result to both Dempster’s Combination Rule and Yager’s Combination Rule it is apparent the difference is in the way evidence is filtered.

Yager’s Combination Rule assigns all conflict to the universal set increasing the numerical value that represents the degree of ignorance within the system. This maintains the evidence. Dempster’s Combination Rule normalizes the conflict out of the system filtering the evidence greatly. Inagaki’s Extra Rule filters the evidence, however looking
at $m(C)$ in figure 5.4 it can be seen that the evidence is scaled by a value determined by both $q(X)$ and $q(\emptyset)$.

$$k = \frac{1}{1 - q(\emptyset) - q(X)}$$

$$m(C) = \left\{ 1 + \left[ \frac{1}{1 - q(\emptyset) - q(X)} \right] q(\emptyset) \right\} q(C)$$

$$= \frac{q(C)[q(X) - 1]}{1 - q(\emptyset) - q(X)}$$

$$m(X) = \left\{ 1 + \left[ \frac{1}{1 - q(\emptyset) - q(X)} \right] q(\emptyset) \right\} q(X) +$$

$$\left\{ 1 + \left[ \frac{1}{1 - q(\emptyset) - q(X)} \right] q(\emptyset) - \frac{1}{1 - q(\emptyset) - q(X)} \right\} q(\emptyset)$$

$$= q(X)$$

**Figure 5.4 Derivation of Inagaki’s Extra Rule**
6. IMPLEMENTATION

Dempster-Shafer is a data fusion/data combination algorithm. It is designed to have multiple inputs from independent sources and generate a single output.

Recall, as previously detailed in the Data Development section of Chapter 2, the data used in this algorithm is taken from the results of Chessa Guilas’ Master’s Thesis. Chessa Guilas created three feature based classifiers: peaks, edges, and corners. Individually the accuracy of the classifiers are 71.2, 58.7, and 50 percent, respectively\(^6\).

In addition to the overall accuracy of the classifier (from this point forward the overall accuracy will be stated as the accuracy based on historical data in order to be more analogous to real world systems), Chessa Guilas determined the individual classifier’s confidence specific to each classification.
6.1 Phase One: Proof of Concept

Phase one of the implementation of Dempster-Shafer’s Theory of Evidence was considered a proof of concept. It was necessary to determine if this type of data fusion was a viable choice for this application.

Given both the individual classifier’s confidence and the corresponding classification two arrays were created for each target, a cell and double array. The cell array can contain an entire string at each index. The feature based classification was placed into this array starting at index one and ending at index ten. The beliefs in the accuracy of each classification were placed into the double array at the corresponding index to the matching classification.

There are sixteen arrays for each feature based classifier. Each input image has been classified three times, once by each classifier (peaks, edges and corners). The classification arrays from each classifier pertaining to the same target are combined.
creating a thirty element array. Where index one refers to the peaks classification, index two to the edges classification, and index three to the corners classification (this repeats for the ten images). The same organization is applied to the probability array. This was done to keep all the data for each target together and reduce the overall number of arrays ideally increasing the efficiency of the algorithm.

Figure 6.2 Diagram of Data Organization

The numerator of Dempster’s Combination Rule is the sum of the intersects and the denominator is one minus the sum of the null sets as shown in equation 6.1.

\[
m^{1,2,3}(D) = \frac{\sum_{A \cap B \cap C = D} m^1(A)m^2(B)m^3(C)}{1 - \sum_{A \cap B \cap C = \emptyset} m^1(A)m^2(B)m^3(C)}
\]  

(6.1)

The basic probability assignments \(m^1(A)\), \(m^2(B)\), and \(m^3(C)\) represent the belief in the classification made by each of the classifiers: peaks, edges, and corners, respectively. The value of \(m^{1,2,3}(D)\) is the belief in the result of the fused data. In order to determine
which classifiers intersect and which make up the null set the data has to be “known.”
The combination rule is dependent on knowledge pertaining to the specifics of which
feature based classifiers agree or disagree, how many agree, and how many disagree.

Using the cell array, which contained the classification, a switch case statement
(analogous to an if statement for standard value types) is used to systematically check for
classifier agreement. The algorithm keeps track of which classifiers in particular agree or
disagree. There are three possibilities, all agree, none agree, or two agree. Within the case
of two agreeing there exist three possibilities (classifier1 with 2, classifier 1 with 3, or
classifier 2 with 3). Knowing which feature based classifiers agree or disagree, it is then
possible to use their corresponding probabilities to calculate the sum of intersects and the
sum of null sets.

Figure 6.3 Diagram of Determination of Classifier Agreement

To calculate these values an assumption needed to be made about the given data.
For the input to the algorithm a probability (a belief) is given. The question is what does
the compliment of the probability/belief mean. Given a belief of 85 percent, does that mean that 15 percent of the time it has to be another target or does it mean that 15 percent of the time the state is unknown- could it be the target or any other target? This assumption greatly effects the value of the intersect and null sets. If the state is unknown than that value can be used in the calculation of the intersect, otherwise the value is used in the calculation of the null. In this algorithm it was chosen to be an unknown state. This guaranteed that no data would be accidentally fabricated.

Using the formula for a three sensor system, shown in equation 6.1, the fused basic probability assignment (belief) was calculated and depending on classifier agreement all possible combinations were compared until the classification with the highest belief was found and output.

![Figure 6.4 Diagram of Dempster’s Combination Rule after Classifier Agreement Determination](image)

**Figure 6.4 Diagram of Dempster’s Combination Rule after Classifier Agreement Determination**
Phase one was implemented with two types of data in order to represent two different scenarios that could occur in the real world. The first scenario implemented the above outlined algorithm using the data received from the feature based classifiers completely unmodified by previous knowledge of the system. This represented a system in which the sensors in use are completely new and untested. The second scenario, the one more likely to be seen in the real world, used data that was scaled based on the historical accuracy of the feature based classifiers. The assumption was that in the real world past training exercises and testing would have been done to determine the accuracy that could be expected from each of the sensors used regardless of type.

6.2 Phase Two: Generalization of Phase One

As stated in the last section, phase one of the implementation was done as a proof of concept. It was determined that Dempster-Shafer’s Theory of Evidence was a viable option. To view the specific results please turn to chapter 7.

Phase two of the implementation was the generalization of the first phase. Due to the fact the phase one was a proof of concept the code was extremely rigid. Essentially, it was designed specifically for the three feature base classifiers created in Chessa Guilas’ Master’s Thesis. The data for each of those classifiers was entered straight into the code. The only flexibility inherent to the phase one algorithm existed due to the fact that all data was written within a data specific MatLab file that was called into the algorithm. Multiple data files could have been written if necessary, but there are very strict bounds on the data. Without adding to the code nothing could be changed. In phase one there were three classifiers, eight targets, and ten images per target. With phase two all parameters just mentioned are completely modifiable without changing any of the code.
For the code to be as dynamic as desired input from the user was required. The algorithm asks the user specifically for the amount of sensors (it was thought that the term sensor was more general than feature based classifier) in the system, the number of targets, and the historical accuracy of the sensor. If the historical accuracy is unknown an input of one simply implies full confidence in the sensors assignment of belief. The user is then asked for the name of the file containing the data.

For the algorithm to perform correctly the data has to be the right file type. An excel worksheet is the expected file type. The placement of the data within the worksheet is inconsequential. The data pertaining to either one sensor or all sensors can be placed within the same file. If data from all sensors exists within the same file, each sensor can be specific to a worksheet or all sensors can exist within a single worksheet. The arbitrary placement of data is possible because MatLab opens the file with the name specified and asks that the user select the data corresponding to a single sensor as shown in figure 6.5.

![Figure 6.5 Example of Data Selection](image)
The row detailing the content of each of the columns does not get selected neither does the empty cells surrounding the table containing the data. Only the cells containing the data need be selected. Selection is executed by highlighting the cells of interest and clicking OK in the dialogue box.

There is an expected format that the data must have. The data has to be split in two columns without a cell between the split, however it is not necessary for the two columns to be of equal length. This prevents any kind of limitation on the number of targets. If there is some reason that the data has to exist in one column the user simply has to double the value input for the number of targets. Between each target set there must be an empty cell. This is used to distinguish between the targets. For each image or to be more general, for each classification attempt there needs to be four columns with no empty cells between. The information found in the four columns are (in this exact order), the true target, the sensor’s belief in its classification, image number (attempt number), and classification.

MatLab places the data selected within two separate arrays. One of the arrays contains only the numerical data while the other contains the text based data. Each of these arrays has a depth corresponding to the number of sensors to allow for easy access of data. The data placement within each of these layers is the same as found in the excel worksheet. MatLab puts placeholders in the cells without numerical data in order to maintain original formatting for the number array and vice versa for the text array.

With the generalization of the data input it became theoretically possible that an infinite number of sensors could be input into the data fusion algorithm. This required the creation of an algorithm that determined and recorded which sensors agreed and which
disagreed for the calculation of the intersect and null sets because it was no longer practical to manually code every possible combination as done in phase one. To keep the processing speed relatively high and the space requirement low, the algorithm compared the classification of each sensor and recoded the index of the classifier as related to the depth within the text array. All agreements were listed in descending rows and disagreements were listed in following columns. An example is detailed below.

Example 6.1: Table 6.1 lists the data that would be input into the algorithm given a system of three sensors and one target. The corresponding index or location of the data for each sensor is displayed next to the sensor title. To be consistent with the content of this thesis the three sensors are the feature based classifiers, peaks, edges, and corners.

*The target is the ZSU.*

<table>
<thead>
<tr>
<th>Peaks (1)</th>
<th>Edges (2)</th>
<th>Corners (3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual tank</td>
<td>Image num</td>
<td>Prob</td>
</tr>
<tr>
<td>[0]ZSU 14931</td>
<td>0.89</td>
<td>ZSU</td>
</tr>
<tr>
<td>[0]ZSU 14999</td>
<td>0.7</td>
<td>ZSU</td>
</tr>
<tr>
<td>[0]ZSU 15065</td>
<td>0.7</td>
<td>ZSU</td>
</tr>
<tr>
<td>[0]ZSU 15066</td>
<td>0.8</td>
<td>T62</td>
</tr>
<tr>
<td>[0]ZSU 15132</td>
<td>0.8</td>
<td>T62</td>
</tr>
<tr>
<td>[0]ZSU 15196</td>
<td>0.7</td>
<td>ZSU</td>
</tr>
<tr>
<td>[0]ZSU 15198</td>
<td>0.8</td>
<td>T62</td>
</tr>
<tr>
<td>[0]ZSU 15261</td>
<td>0.89</td>
<td>ZSU</td>
</tr>
<tr>
<td>[0]ZSU 15265</td>
<td>0.89</td>
<td>ZSU</td>
</tr>
<tr>
<td>[0]ZSU 15267</td>
<td>0.89</td>
<td>ZSU</td>
</tr>
</tbody>
</table>
For the agreement algorithm the only data of interest is the classification of each of the feature based classifiers. The classifications are placed in table 6.2 to make comparison simpler.

Table 6.2 Example of Classification Agreement and Disagreement

<table>
<thead>
<tr>
<th>Classification</th>
<th>Peaks (1)</th>
<th>Edges (2)</th>
<th>Corners (3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classification #1</td>
<td>ZSU</td>
<td>ZSU</td>
<td>ZSU</td>
</tr>
<tr>
<td>Classification #2</td>
<td>ZSU</td>
<td>ZSU</td>
<td>ZS1</td>
</tr>
<tr>
<td>Classification #3</td>
<td>ZSU</td>
<td>ZSU</td>
<td>ZS1</td>
</tr>
<tr>
<td>Classification #4</td>
<td>T62</td>
<td>D7</td>
<td>BDRM2</td>
</tr>
<tr>
<td>Classification #5</td>
<td>T62</td>
<td>ZSU</td>
<td>T62</td>
</tr>
<tr>
<td>Classification #6</td>
<td>ZSU</td>
<td>ZSU</td>
<td>T62</td>
</tr>
<tr>
<td>Classification #7</td>
<td>T62</td>
<td>ZSU</td>
<td>ZS1</td>
</tr>
<tr>
<td>Classification #8</td>
<td>ZSU</td>
<td>T62</td>
<td>T62</td>
</tr>
<tr>
<td>Classification #9</td>
<td>ZSU</td>
<td>ZSU</td>
<td>BDRM2</td>
</tr>
<tr>
<td>Classification #10</td>
<td>ZSU</td>
<td>ZSU</td>
<td>ZSU</td>
</tr>
</tbody>
</table>

The output of the agreement algorithm is shown in table 6.3. For each classification attempt, or image in this case, the algorithm outputs an array where each column is a differing target classification and the rows in each column contain the index of the classifier that agrees with that particular classification.

Table 6.3 Example of Output of Agreement Algorithm

<table>
<thead>
<tr>
<th>Classification Attempt #1</th>
<th>Classification Attempt #6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Classification Attempt #2</td>
<td>Classification Attempt #7</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Classification Attempt #3</td>
<td>Classification Attempt #8</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Classification Attempt #4</td>
<td>Classification Attempt #9</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Classification Attempt #5</td>
<td>Classification Attempt #10</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>
In table 6.3 the fourth classification attempt shows that none of the classifiers agree, conversely the first and last classification attempt show all classifiers agreeing. Recall that the corresponding target classifications can be seen in table 6.2 above.

The array created by the agreement algorithm is then used to calculate the intersect and null sets. Recall equation 6.1, the numerator is the sum of all intersections and the denominator includes the sum of all null sets. The algorithm needs to calculate the permutation of every column from the agreement algorithm. Before delving into the calculation of the permutations let it be noted that the sum of the null set is simply the multiplication of the disagreeing classifier’s belief in their classifications. The value does not differ regardless of which target (column) becomes the focus of the combination algorithm for each classification attempt.

As previously stated, there can exist an infinite amount of sensors in the system. Given a binary system the number of permutations increases at a rate of two to the number of sensors that agree. In phase one and two there is a binary relationship represented by the belief given from the sensor, either it is the target or it can be any target. This type of exponential increase lent itself well to a truth table format. What would normally have been a one was replaced with the classifier’s belief in its classification and the zero became the compliment to the value replacing the one. Each column of the truth table is specific to a different sensor. In terms of the classifiers that disagree with the current classification focus, the compliment of the classifier’s belief in its classification had to be concatenated to the end of each of the rows of the truth table as that value represents an intersecting point. All columns of each row were multiplied
together and the resulting values summed to obtain the sum of all intersections. Example 6.2 is a continuation of example 6.1 and shows the above algorithm progression.

Example 6.2: Recall example 6.1 and the correspond tables. Looking at table 6.3 the most classifiers in agreement for any particular classification attempt is three. At most there are eight different intersection values. In this example classification attempts one, four, and eight will be fully explored.

Classification Attempt #1:

In this particular case all classifiers agree so there is only one focus. The target classification focus is ZSU. At this point the end result of any combination rule is obvious, however for the sake of robustness even the case was calculated within the algorithm. Each of the three classifiers has a different belief in its classification as shown in equation 6.2. The compliments of those values are shown in equation 6.3.

\[
m^1(A) = 0.89, \quad m^2(B) = 0.72, \quad m^3(C) = 0.63
\]

\[
1 - m^1(A) = 0.11, \quad 1 - m^2(B) = 0.28, \quad 1 - m^3(C) = 0.37
\]

In the equations above the superscripts 1, 2, and 3 represent the three different classifiers, peaks, edges and corners, respectively. A truth table is created using the values above, table 6.4, to calculate the sum of intersections.
Table 6.4 Permutation Connected to Classification Attempt #1

<table>
<thead>
<tr>
<th>0.89</th>
<th>0.72</th>
<th>0.63</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.89</td>
<td>0.72</td>
<td>0.37</td>
</tr>
<tr>
<td>0.89</td>
<td>0.28</td>
<td>0.63</td>
</tr>
<tr>
<td>0.89</td>
<td>0.28</td>
<td>0.37</td>
</tr>
<tr>
<td>0.11</td>
<td>0.72</td>
<td>0.63</td>
</tr>
<tr>
<td>0.11</td>
<td>0.72</td>
<td>0.37</td>
</tr>
<tr>
<td>0.11</td>
<td>0.28</td>
<td>0.63</td>
</tr>
<tr>
<td>0.11</td>
<td>0.28</td>
<td>0.37</td>
</tr>
</tbody>
</table>

For each row every column is multiplied to calculate the value or mass of the intersection.

Table 6.5 Masses of Intersection for Classification Attempt #1

<table>
<thead>
<tr>
<th>0.403704</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.237096</td>
</tr>
<tr>
<td>0.156996</td>
</tr>
<tr>
<td>0.092204</td>
</tr>
<tr>
<td>0.049896</td>
</tr>
<tr>
<td>0.029304</td>
</tr>
<tr>
<td>0.019404</td>
</tr>
<tr>
<td>0.011396</td>
</tr>
</tbody>
</table>

The values in the above table are then summed and the value generated is the summation of all intersects and the numerator of Dempster’s Combination Rule.

Classification Attempt #4:

All classifiers disagree on the target identity. In some cases Dempster’s Combination Rule can not be used in this particular situation. What determines whether or not Dempster’s Combination Rule can be used is the initial assumption. Is the compliment of the classifier’s belief in its classification the belief that it is not the target identified or the belief that it could be any target including the one identified? If it is the belief that it is not the identified target then there are no intersections and Dempster’s Combination can not be used. In this system the later version
of the compliment was chosen and so this is a situation where Dempster’s
Combination Rule can be used. Equations 6.4 and 6.5 show both the belief
in classification and its compliment, respectively.

\[ m^1(A) = 0.80, m^2(B) = 0.80, m^3(C) = 0.68 \] (6.4)

\[ 1 - m^1(A) = 0.20, 1 - m^2(B) = 0.20, 1 - m^3(C) = 0.32 \] (6.5)

In this case there is only one classifier in agreement at a time so the
number of permutations is two. However, unlike classification attempt
number one there are three different target focuses. According to the
classifiers the target could be the T62, the D7, or the BDRM2. This
classification attempt is going to require three different truth tables
(Figure 6.6) so that the three different summations of the intersects can be
calculated.

\[
\begin{array}{ccc}
0.80 & 0.80 & 0.68 \\
0.20 & 0.20 & 0.32 \\
\end{array}
\]

Figure 6.6 Permutation Connected to Classification Attempt #4

The compliments of the disagreeing classifiers have to be concatenated to
the rows of the truth table in order for the intersection masses to be
calculated.

\[
\begin{array}{ccc}
0.80 & 0.20 & 0.32 \\
0.20 & 0.20 & 0.32 \\
\end{array}
\]

\[
\begin{array}{ccc}
0.80 & 0.20 & 0.32 \\
0.20 & 0.20 & 0.32 \\
0.68 & 0.20 & 0.20 \\
\end{array}
\]

Figure 6.7 Permutation Connected to Classification Attempt #4 with Concatenated Disagreeing Classifier Belief Compliments

From the truth tables in figure 6.7 it is possible to calculate the mass of
the intersects. The values are shown in figure 6.8.

\[
\begin{array}{ccc}
0.0512 & 0.0512 & 0.0272 \\
0.0128 & 0.0128 & 0.0128 \\
\end{array}
\]

Figure 6.8 Masses of Intersection for Classification Attempt # 4
Once again, from the masses above the three different values for the summation of intersects can be calculated and used in Dempster’s Combination Rule.

Classification Attempt #8:

Both extremes have already been examined. Classification attempt number eight is a scenario that is more likely to occur. Two of the three classifiers agree. While it is necessary to demonstrate the application of this algorithm on a standard scenario, an interesting side note is that the one disagreeing classifier is the most accurate of all classifiers. Equations 6.6 and 6.7 show both the belief in classification and its compliment, respectively.

\begin{align*}
m^1(A) &= 0.89, \quad m^2(B) = 0.72, \quad m^3(C) = 0.80 \\
1 - m^1(A) &= 0.11, \quad 1 - m^2(B) = 0.28, \quad 1 - m^3(C) = 0.20
\end{align*} \quad (6.6) \quad (6.7)

In this case there are two classifiers in agreement so the number of permutations is four. According to the classifiers the target could be either the ZSU or the T62. This classification attempt is going to require two different truth tables (Figure 6.9) so that the two different summations of the intersects can be calculated.

\begin{tabular}{ccc}
0.72 & 0.80 & 0.89 \\
0.72 & 0.80 & 0.11 \\
0.28 & 0.20 & 0.20 \\
0.28 & 0.20 & 0.20
\end{tabular}

\textbf{Figure 6.9 Permutation Connected to Classification Attempt #8}

The compliments of the disagreeing classifiers have to be concatenated to the rows of the truth table in order for the intersection masses to be calculated.
From the truth tables in figure 6.10 it is possible to calculate the mass of
the intersects. The values are shown in figure 6.11.

From the masses above the two different values for the summation of
intersects can be calculated and used in Dempster’s Combination Rule.

With the summation of the intersections and the summation of the null sets
implemented, equation 6.1 was calculated for each of the target focuses. As was noted
earlier the change of target focus does not affect the value calculated for the summation
of the null set. In other words the denominator of Dempster’s Combination Rule does not
change in this application. The classification through data fusion can be done completely
based on the summation of the intersects. This value is also known as the ground
probability assignment. The true ramifications of this are shown in the results and
conclusions chapters. A simple block diagram of the algorithm is shown in figure 6.12
below.
6.3 Phase Three: Inagaki’s Unified Combination Rule

Between phase two and three there were two changes done to the algorithm detailed in the previous section. The first change was the change of the combination rule.
Phase two implemented Dempster-Shafer’s Theory of Evidence using Dempster’s Combination Rule. Phase three used Inagaki’s Unified Combination Rule. The other change was the way the classifier’s belief in its own accuracy was calculated.

The main goal of implementing Inagaki’s Unified Combination Rule was to increase the accuracy further than that seen in phase one and phase two. The original thought was that through adjustment to \( k \), please see equation 6.8 for reference\(^7\), the increase in accuracy would be obtained.

\[
0 \leq k \leq \frac{1}{1 - q(\emptyset) - q(X)}
\]

(6.8)

In the previous section it was mentioned that the denominator of Dempster’s Combination Rule does not change with the change of target focus. The same applies to Inagaki’s Unified Combination Rule. This lack of change meant that Inagaki’s Unified Combination Rule would classify targets in the same manner regardless of the value of \( k \). Every combination rule shown in figure 6.13 would have the same result.

<table>
<thead>
<tr>
<th>Yager’s Rule</th>
<th>Dempster-Shafer’s Rule</th>
<th>Inagaki’s Extra Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>( k )</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
&\frac{1}{1 - q(\emptyset)} \\
&\frac{1}{1 - q(\emptyset) - q(X)} \\
&0
\end{align*}
\]

Figure 6.13 Rule Selection Dependent on \( k \) Value

For this reason the way the classifier’s belief in its own accuracy was calculated was changed. The three values that were needed were the belief that it was the target, the belief that it was not the target, and the belief of ignorance. These are denoted \( m(A) \),
$m(\sim A)$, and $m(X)$, respectively. These three values were calculated using the historical accuracy of the classifiers, which will be denoted $HA$ in the following equations for simplicity. Equation 6.9, 6.10 and 6.11 show the method used to calculate the three values.

$$m(A) = (Belief \ supplied \ by \ Classifier)xAH$$  \hspace{1cm} (6.9)

When calculating the belief that it was specifically not the target there was no way to justify that the ignorance of the system scaled by the historical accuracy would equate to that representative value. However, that value scaled by the number of targets could theoretically represent $m(\sim A)$.

$$m(\sim A) = \frac{[1 - (Belief \ supplied \ by \ Classifier)]xAH}{(Number \ of \ Targets)}$$  \hspace{1cm} (6.10)

The scaling of $m(\sim A)$ by the number of targets also increased the ignorance of the system which firmly justified the above assumptions.

$$m(X) = 1 - [m(A) + m(\sim A)]$$  \hspace{1cm} (6.11)

Once the above values were calculated within the algorithm the ground probability assignment (the summation of the intersects) and the summation of the null sets needed to be calculated just as in the last section. In replacement of Dempster’s Combination Rule Inagaki’s Combination Rule was used. Recall, Inagaki’s Combination Rule shown in equation 6.12.

$$m(C) = \left[1 + k \ q(\emptyset)\right]q(C), \ C \neq X, \ \emptyset$$  \hspace{1cm} (6.12)

The algorithm to calculate the ground probability assignment changed slightly. However, the idea remained the same. Given all beliefs corresponding with agreement create a truth table to generate all permutations. Use the truth table to calculate the ground probability assignment. The rest of the algorithm was unchanged.
7. RESULTS

7.1 Phase One and Two

The purpose of this thesis was to explore the applicability of Dempster Shafer’s Theory of Evidence to automatic target recognition. The first phase was a proof of concept as explained in the previous chapter. In order to continue with this avenue of combination the resultant accuracy of the system had to be greater than that obtained by the best feature classifier independently. The peaks feature classifier is most accurate.

<table>
<thead>
<tr>
<th></th>
<th>ZSU</th>
<th>ZIL</th>
<th>D7</th>
<th>2S1</th>
<th>SLICY</th>
<th>BDRM2</th>
<th>BTR60</th>
<th>T62</th>
<th>Error of Exclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZSU</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>30%</td>
</tr>
<tr>
<td>ZIL</td>
<td>1</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10%</td>
</tr>
<tr>
<td>D7</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>60%</td>
</tr>
<tr>
<td>2S1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>20%</td>
</tr>
<tr>
<td>SLICY</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>BDRM2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>4</td>
<td>50%</td>
</tr>
<tr>
<td>BTR60</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>60%</td>
</tr>
<tr>
<td>T62</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0%</td>
</tr>
</tbody>
</table>

Error of Inclusion: 46.15% 10% 0% 0% 9.09% 44.44% 0% 52.38% Accuracy: 71.25%

Figure 7.1 Confusion Matrix for the Peaks Feature Classifier
Figure 7.1 is the confusion matrix for the peaks feature classifier. It set the threshold that had to be out performed. From the figure it can be seen that the accuracy necessary to overcome was 71.25 percent.

The proof of concept was limited to overall accuracy. Error of exclusion and error of inclusion were not used in any way to justify the research into this area of combination. However, an increase in accuracy would invariably lead to a change in the error as can be seen be comparing figures 7.1 and 7.2. Where figure 7.2 is the resultant confusion matrix when using Dempster-Shafer’s Theory of Evidence with Dempster’s Combination Rule.

![Confusion Matrix](image)

Figure 7.2 shows an accuracy that is greater than that of the peaks feature classifier as expressed in figure 7.1.

The confusion matrix in figure 7.2 is the result of using Dempster-Shafer’s Theory of Evidence given un-weighted data. The assumption was that there existed no knowledge of the overall accuracy of the sensors (feature based classifiers as termed within this thesis) within the system. The only data seen was the confidence of recognition expressed by the classifier itself. For example, the system is blind to the peaks classifier having an accuracy of 71.25 percent historically. The system only has
access to the data that the peaks classifier has a 90 percent confidence that the target has been accurately identified.

The results obtained proved that this was a viable area of research for the combination of the feature classifiers, however the accuracy obtained was not as high as was thought could be obtained with Dempster-Shafer’s Theory of Evidence. All information known about the system had not been utilized. Figure 7.3 is the confusion matrix of the system given knowledge of the historical accuracy of the feature classifiers.

<table>
<thead>
<tr>
<th></th>
<th>ZSU</th>
<th>ZIL</th>
<th>D7</th>
<th>2S1</th>
<th>SLICY</th>
<th>BDRM2</th>
<th>BTR60</th>
<th>T62</th>
<th>Error of Exclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZSU</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>30%</td>
</tr>
<tr>
<td>ZIL</td>
<td>0</td>
<td>9</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10%</td>
</tr>
<tr>
<td>D7</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>30%</td>
</tr>
<tr>
<td>2S1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>10%</td>
</tr>
<tr>
<td>SLICY</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>BDRM2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>40%</td>
</tr>
<tr>
<td>BTR60</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>60%</td>
</tr>
<tr>
<td>T62</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>10</td>
<td>0%</td>
</tr>
<tr>
<td>Error of Inclusion</td>
<td>12.50%</td>
<td>10.00%</td>
<td>12.50%</td>
<td>0.00%</td>
<td>9.09%</td>
<td>40.00%</td>
<td>0.00%</td>
<td>50.00%</td>
<td>77.50%</td>
</tr>
</tbody>
</table>

Figure 7.3 Confusion Matrix for Weighted System

The confidence values supplied by the feature classifiers are weighted by the historical accuracies and then combined using Dempster Shafer’s Theory of Evidence. As can be seen in figure 7.3 the accuracy of the system increases 2.5 percent.

7.2 Phase Three

When using all of the data available the greatest accuracy obtained using Dempster-Shafer’s Theory of Evidence was 77.50 percent. While it was greater than the accuracy achieved with the un-weighted system it still was under what was desired. In order to create a system more accurate other combination rules were used in conjunction with Dempster-Shafer’s Theory of Evidence. Specifically, Inagaki’s Unified
Combination Rule due to the fact that the treatment of conflict was adjustable. It was hypothesized that this adjustability would lead to that desired increase in accuracy.

Figure 7.4 shows that the hypothesis was proved wrong. The accuracy obtained using Inagak’s Unified Combination Rule was 75 percent. Recall that is the same as the un-weighted system. The variable treatment of the conflict had no effect. The result was less than that obtained by the weighted system using Dempster’s Combination Rule. In the next chapter various reasons for this are detailed.

<table>
<thead>
<tr>
<th></th>
<th>ZSU</th>
<th>ZIL</th>
<th>D7</th>
<th>2S1</th>
<th>SLICY</th>
<th>BDRM2</th>
<th>BTR60</th>
<th>T62</th>
<th>Error of Exclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZSU</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>30%</td>
</tr>
<tr>
<td>ZIL</td>
<td>0</td>
<td>9</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10%</td>
</tr>
<tr>
<td>D7</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>40%</td>
</tr>
<tr>
<td>2S1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>10%</td>
</tr>
<tr>
<td>SLICY</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>BDRM2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>4</td>
<td>50%</td>
</tr>
<tr>
<td>BTR60</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>60%</td>
</tr>
<tr>
<td>T62</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0%</td>
</tr>
<tr>
<td>Error of Inclusion</td>
<td>0.00%</td>
<td>10.00%</td>
<td>14.29%</td>
<td>18.18%</td>
<td>9.09%</td>
<td>44.44%</td>
<td>0.00%</td>
<td>52.38%</td>
<td>Accuracy: 75.00%</td>
</tr>
</tbody>
</table>

Figure 7.4 Confusion Matrix using Inagaki’s Combination Rule

7.3 Combination Rules Compared to Chessa Guilas’ Results

Upon discovering that Dempster-Shafer’s Theory of Evidence could indeed be applied to the fusion of feature based classifiers, it was necessary to understand how it compared to classical probability combination. To that end the results of this thesis are compared to the results of Chessa Guilas’ thesis. The method of combination used in Guilas’ thesis can be review in chapter 2 section 4.4.

Guilas’ desired accuracy as high as possible; to that end the confidence of each feature based classifier was scaled by an initial value determined by viewing the probability functions of each feature for each target. The values arrived at are shown in figure 7.5.
Using these values the data was combined and the confusion matrix in figure 7.6 was the result. The system accuracy was 82.5 percent which is a full 5 percent greater than the highest achieved by Dempster-Shafer’s Theory of Evidence.

Guilas also ran her combination algorithm using two other weighting schemes. The first of those schemes is shown in figure 7.7, it is global weighting. Regardless of the target the same weight is applied to the confidence supplied by the feature based classifier as determined solely based on the classifier itself. In other words if the confidence value is reported from the peaks classifier it would be weighted by a value of 40 percent, likewise both values from edges and corners would be weighted by a value of 30 percent. The second scheme was obtained through an iterative process. The values listed in figure 7.9 were found to give the highest possible system accuracy. The accuracies of the global weighting and iterative weighting schemes can be seen in figures 7.8 and 7.10, respectively. Through global weighting automatic target recognition was accurate 85 percent of the time. The highest accuracy obtained by Chessa Guilas using...
the iterative process was 86.25 percent. Both cases out perform the result of the initial weights and Dempster-Shafer’s Theory of Evidence.

<table>
<thead>
<tr>
<th></th>
<th>ZSU</th>
<th>ZIL</th>
<th>D7</th>
<th>2S1</th>
<th>SLICY</th>
<th>BDRM2</th>
<th>BTR60</th>
<th>T62</th>
<th>Error of Exclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edges</td>
<td>30%</td>
<td>30%</td>
<td>30%</td>
<td>30%</td>
<td>30%</td>
<td>30%</td>
<td>30%</td>
<td>30%</td>
<td>2</td>
</tr>
<tr>
<td>Corners</td>
<td>30%</td>
<td>30%</td>
<td>30%</td>
<td>30%</td>
<td>30%</td>
<td>30%</td>
<td>30%</td>
<td>30%</td>
<td>1</td>
</tr>
<tr>
<td>Peaks</td>
<td>40%</td>
<td>40%</td>
<td>40%</td>
<td>40%</td>
<td>40%</td>
<td>40%</td>
<td>40%</td>
<td>40%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Figure 7.7 Global Weight Scheme\(^{(6)}\)

<table>
<thead>
<tr>
<th></th>
<th>ZSU</th>
<th>ZIL</th>
<th>D7</th>
<th>2S1</th>
<th>SLICY</th>
<th>BDRM2</th>
<th>BTR60</th>
<th>T62</th>
<th>Error of Exclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZSU</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>80%</td>
</tr>
<tr>
<td>ZIL</td>
<td>1</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>80%</td>
</tr>
<tr>
<td>D7</td>
<td>2</td>
<td>0</td>
<td>7</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>70%</td>
</tr>
<tr>
<td>2S1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>SLICY</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>BDRM2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>1</td>
<td>80%</td>
</tr>
<tr>
<td>BTR60</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>7</td>
<td>1</td>
<td>70%</td>
</tr>
<tr>
<td>T62</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>100%</td>
</tr>
</tbody>
</table>

Figure 7.8 Confusion Matrix as a Result of Global Weighting\(^{(6)}\)

<table>
<thead>
<tr>
<th></th>
<th>ZSU</th>
<th>ZIL</th>
<th>D7</th>
<th>2S1</th>
<th>SLICY</th>
<th>BDRM2</th>
<th>BTR60</th>
<th>T62</th>
<th>Error of Exclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edges</td>
<td>40%</td>
<td>30%</td>
<td>40%</td>
<td>30%</td>
<td>40%</td>
<td>30%</td>
<td>30%</td>
<td>30%</td>
<td>2</td>
</tr>
<tr>
<td>Corners</td>
<td>30%</td>
<td>30%</td>
<td>30%</td>
<td>30%</td>
<td>40%</td>
<td>30%</td>
<td>30%</td>
<td>30%</td>
<td>1</td>
</tr>
<tr>
<td>Peaks</td>
<td>30%</td>
<td>40%</td>
<td>30%</td>
<td>30%</td>
<td>30%</td>
<td>40%</td>
<td>40%</td>
<td>40%</td>
<td>20%</td>
</tr>
</tbody>
</table>

Figure 7.9 Weights Determine through Iterative Process\(^{(6)}\)

<table>
<thead>
<tr>
<th></th>
<th>ZSU</th>
<th>ZIL</th>
<th>D7</th>
<th>2S1</th>
<th>SLICY</th>
<th>BDRM2</th>
<th>BTR60</th>
<th>T62</th>
<th>Error of Exclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZSU</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>20%</td>
</tr>
<tr>
<td>ZIL</td>
<td>1</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>20%</td>
</tr>
<tr>
<td>D7</td>
<td>1</td>
<td>0</td>
<td>8</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20%</td>
</tr>
<tr>
<td>2S1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>SLICY</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>BDRM2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>1</td>
<td>20%</td>
</tr>
<tr>
<td>BTR60</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>7</td>
<td>1</td>
<td>30%</td>
</tr>
<tr>
<td>T62</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0%</td>
</tr>
</tbody>
</table>

Figure 7.10 Confusion Matrix as a Result of Weights Determine through Iterative Process\(^{(6)}\)

Conclusions in regards to the performance difference seen between Dempster–Shafer’s Theory of Evidence and classical probability will be discussed in the next chapter.
8. CONCLUSIONS AND FUTURE WORK

Dempster-Shafer’s Theory of Evidence was implemented with two goals. The first goal was to discover if it was applicable to the automatic target recognition research being done at Calpoly. The second goal was contingent on first, if Dempster-Shafer was found to be applicable determine how its results compare to classical probability theory.

8.1 Dempster-Shafer as Compared to Results found by Chessa Guilas

In the last chapter the results of Dempster-Shafer’s Theory of Evidence are shown as are the results from Chessa Guilas’ thesis. It is evident that Chessa Guilas’ results are far better than those found by this thesis. It can be said that in this particular application classical probability out performs Dempster-Shafer’s Theory of Evidence. There are multiple reasons for this outcome.

Dempster-Shafer Theory of Evidence demands that all sources of evidence be independent\(^{[3]}\). In this case, all feature classifiers have to be independent. The feature based classifiers are not independent. There was one sensor that was used, Synthetic
Aperture Radar, and from the image produced these classifications were developed. The most that can be said is that the classifiers are quasi-independent, in that the features used are not directly related.

More importantly the theory is designed to use different kinds of evidence to adjust the belief in each possible hypothesis. For example, consider that the set of targets from the MSTAR database were in motion (those that can be) and a sensor that determines speed was used. The data supplied by the speed sensor would have increased the belief in specific targets, decreased the belief in others, and simple thrown out those that the speed reading could not have applied to. The data from the speed sensor would have been used in conjunction with the feature based classifiers through the Dempster-Shafer Theory of Evidence resulting in a single classification with one algorithm. The data from the speed sensor could be used in classical probability but not in the same way. Had this been the system under test it is thought that this thesis would have shown a more efficient combination algorithm. The accuracy of the classical probability combination would still have been higher.

The main reason for the discrepancy in accuracy lay in the weight scaling. All phases of the Dempster-Shafer algorithm relied on either no information other than that supplied by the classifier itself or a weighted version of that data scaled due to the credibility of the classifier based on historical data. Examples of both scenarios can be found in the real world. The algorithm implementing the classical probability combination weighted the data based on knowledge specific to the exact situation. The weighting was not based on the historical accuracy of the classifier rather it was based on
educated guessing and later on iterative processing. Dempster-Shafer traded accuracy for robustness.

8.2 Reasons for the Results of Different Combination Rules

After implementing all three phases of the algorithm three separate combination rules had been used with Dempster-Shafer’s Theory of Evidence. The three combination rules were Dempster’s Combination Rule, Yager’s Combination Rule and Inagaki’s Unified Combination Rule. The hypothesis was that each of the rules would have a different result and Inagaki’s Unified Combination Rule would lead to the best possible accuracy obtainable.

8.2.1 Lack of Results for Yager’s Combination Rule

In the results chapter there is not a confusion matrix that is specific to Yager’s Combination Rule. This is due to the fact that only one type of evidence was used to recognize the targets and that the classifiers were only quasi-independent. The results for Yager’s Combination Rule are the same as those found using Dempster’s Combination Rule.

The only difference between Dempster’s Combination Rule and Yager’s Combination Rule is in the treatment of conflict. Dempster’s Combination Rule uses a normalization factor to completely ignore the conflict. Yager’s Combination Rule adds the conflict to the universal set showing an increase in ignorance of the system. The normalization factor is defined in chapter 3 as $1-k$, where k is the mass of the null set. For each of the possible classifications presented by the feature based classifiers the mass of the null set will always be the same due to the single source of evidence. Without change to the null set the denominator no longer affects the relative masses for each classification possibility. The normalization factor can be ignored in this case. With the denominator of
Dempster’s Combination Rule gone, Dempster’s Combination Rule becomes Yager’s Combination Rule.

8.2.2 Inagaki’s Unified Combination Rule

Inagaki’s Unified Combination Rule needed three different probabilities to be implemented. For Dempster’s Combination there were only two different probabilities, the probability or belief that the classifier was certain in its classification and the compliment to that value, the belief that it could be anything else. Inagaki’s Combination rule needed a third, the probability that the target was not what the classifier had classified it as. In chapter 6 the way this third probability is created is fully detailed.

The creation of this third probability using the credibility of the classifier is the reason that the highest obtainable accuracy was equal to the results of Dempster’s Combination Rule given non-weighted data. Even though all data about the classifiers was utilized the creation of the third probability created a system where the input data was equivalent to the non-weighted data input into Dempster’s Combination Rule. In other words the creation of the third probability nullified the weighting affect of knowing the credibility of the feature based classifiers.

8.3 Future Work

The final algorithm created is completely general. The input can be as many classifiers or types of evidence as desired as long as the data is formatted in the correct way as detailed in the chapter 6. It was tested to make sure that there were no bugs with more than the three classifiers presented within this thesis. However, the true effect of adding more classifiers to the algorithm was not explored. How much higher can the accuracy of the overall system be increased above the accuracy of the most accurate classifier in the system? This question could be investigated.
Furthermore, there exists many other combination rules that can be used with the framework devised by Glen Shafer and those other combination rules could be implemented to determine which combination rule is truly the best for this application. However, until there are other sources of evidence the true potential of Dempster-Shafer's Theory of Evidence will not be realized. Depending on the future work involved with some of the other research areas there is potential that the automatic target recognition algorithms can be implemented on a moving target where Dempster-Shafer is used for final classification to determine whether the classified target needs to be tracked. At which point the tracking algorithms previously and currently being created could be used. That particular thesis would tie together all areas of research overseen by Professor Saghri.
REFERENCES:


Information Sciences 41: 93-137. 1987a.


APPENDIX A: ALGORITHM CODE

Phase One:

File: wrap_around.m

function wrap_around

%% Information Pertaining to Function
%% The file is a function used as part of the implementation of 
%% phase one of Dempster-Shafer's Theory of Evidence. 
%% This function is the wrap around file. All necessary function calls 
%% are made from this m file. 
%%
%% Created by Scott Seims 
%%
%%
%% PURPOSE: 
%% --- The purpose of this function is to call all necessary functions to 
%% implement Dempster-Shafer's Theory of Evidence. 
%%
%% NOTE: 
%% --- This code is very rigid and it is not recommended that any thing be 
%% built from it unless using it for a proof of concept. 
%%
%% INPUTS: 
%% --- The input to this function is the data.m file. 
%%
end
%OUTPUTS:
%---The output of this function are the data fused classifications in a
% table easily translated to a confusion matrix.
%
%%% Wrap Around File

%Input call for data
data

%Organization of data
[sens_clas_ZSU sens_prob_ZSU] = sens_array_comb(peaks_clas_ZSU,peaks_prob_ZSU,edge_clas_ZSU,...
    edge_prob_ZSU,corner_clas_ZSU,corner_prob_ZSU);
[sens_clas_ZIL sens_prob_ZIL] = sens_array_comb(peaks_clas_ZIL,peaks_prob_ZIL,edge_clas_ZIL,...
    edge_prob_ZIL,corner_clas_ZIL,corner_prob_ZIL);
[sens_clas_D7 sens_prob_D7] = sens_array_comb(peaks_clas_D7,peaks_prob_D7,edge_clas_D7,...
    edge_prob_D7,corner_clas_D7,corner_prob_D7);
[sens_clas_2S1 sens_prob_2S1] = sens_array_comb(peaks_clas_2S1,peaks_prob_2S1,edge_clas_2S1,...
    edge_prob_2S1,corner_clas_2S1,corner_prob_2S1);
[sens_clas_SLICY sens_prob_SLICY] = sens_array_comb(peaks_clas_SLICY,peaks_prob_SLICY,edge_clas_SLICY,...
    edge_prob_SLICY,corner_clas_SLICY,corner_prob_SLICY);
[sens_clas_BDRM2 sens_prob_BDRM2] = sens_array_comb(peaks_clas_BDRM2,peaks_prob_BDRM2,edge_clas_BDRM2,...
    edge_prob_BDRM2,corner_clas_BDRM2,corner_prob_BDRM2);
[sens_clas_BTR60 sens_prob_BTR60] = sens_array_comb(peaks_clas_BTR60,peaks_prob_BTR60,edge_clas_BTR60,...
    edge_prob_BTR60,corner_clas_BTR60,corner_prob_BTR60);
[sens_clas_T62 sens_prob_T62] = sens_array_comb(peaks_clas_T62,peaks_prob_T62,edge_clas_T62,...
    edge_prob_T62,corner_clas_T62,corner_prob_T62);

%Calculation of Dempster-Shafer's Theory of Probability
[dem_clas_ZSU dem_prob_ZSU] = agree(sens_clas_ZSU, sens_prob_ZSU);
[dem_clas_ZIL dem_prob_ZIL] = agree(sens_clas_ZIL, sens_prob_ZIL);
[dem_clas_D7 dem_prob_D7] = agree(sens_clas_D7, sens_prob_D7);
[dem_clas_2S1 dem_prob_2S1] = agree(sens_clas_2S1, sens_prob_2S1);
[dem_clas_SLICY dem_prob_SLICY] = agree(sens_clas_SLICY, sens_prob_SLICY);
[dem_clas_BDRM2 dem_prob_BDRM2] = agree(sens_clas_BDRM2, sens_prob_BDRM2);
[dem_clas_BTR60 dem_prob_BTR60] = agree(sens_clas_BTR60, sens_prob_BTR60);
[dem_clas_T62 dem_prob_T62] = agree(sens_clas_T62, sens_prob_T62);

%Output of classification in easy to read manner
table(:,1) = dem_clas_ZSU;'
table(:,2) = dem_clas_ZIL;'
table(:,3) = dem_clas_D7;'
table(:,4) = dem_clas_2S1;'
table(:,5) = dem_clas_SLICY;'
table(:,6) = dem_clas_BDRM2;'
table(:,7) = dem_clas_BTR60;'
table(:,8) = dem_clas_T62;

table
end
% Information Pertaining to Function
% The file is a file used as part of the implementation of
% phase one of Dempster-Shafer's Theory of Evidence.
% wrap_around.m is the wrap around file. All necessary function calls
% are made from that m file.
%
% Created by Scott Seims
%
% PURPOSE:
% --- All data for the three feature based classifiers were input into this
% file
%
% NOTE:
% --- The classification is placed into a cell array. The basic probability
% assignment
%
% INPUTS:
% --- There is no input to this file, all data is hard coded.
%
% OUTPUTS:
% --- The wrap around function calls this file to have access to the data
% there are no specific outputs.
%
%% Peaks Data
peaks_clas_ZSU = cellstr(’ZSU ’;’ZSU ’;’ZSU ’;’T62 ’;’T62 ’;’ZSU ’;’T62 ’;’ZSU ’;’ZSU ’;’T62 ’;’T62 ’;’ZSU ’;’T62 ’;’T62 ’;’ZSU ’;’T62 ’;’T62 ’;’ZSU ’;’T62 ’);
peaks_prob_ZSU = [0.89,0.7,0.8,0.8,0.7,0.8,0.89,0.89,0.89];
peaks_clas_ZIL = cellstr(’ZIL ’;’ZIL ’;’ZIL ’;’ZIL ’;’ZSU ’;’ZIL ’;’ZIL ’;’ZIL ’;’ZIL ’;’ZIL ’;’ZIL ’;’ZIL ’;’ZIL ’);
peaks_prob_ZIL = [0.75,0.85,0.85,0.35,0.85,0.85,0.85,0.85,0.85,0.85];
peaks_clas_D7 = cellstr(’D7 ’;’T62 ’;’D7 ’;’ZSU ’;’SLICY’;’ZSU ’;’ZSU ’;’D7 ’;’D7 ’;’ZSU ’;’D7 ’;’D7 ’;’ZSU ’);
peaks_prob_D7 = [0.58,0.8,0.65,0.35,0.9,0.35,0.35,0.65,0.65,0.35];
peaks_clas_2S1 = cellstr(’2S1 ’;’2S1 ’;’T62 ’;’2S1 ’;’2S1 ’;’2S1 ’;’2S1 ’;’2S1 ’;’2S1 ’);
peaks_prob_2S1 = [0.8,0.8,0.8,0.8,0.8,0.8,0.8,0.8,0.8,0.8];
peaks_clas_SLICY = cellstr(’SLICY’;’SLICY’;’SLICY’;’SLICY’;’SLICY’;’SLICY’;’SLICY’;’SLICY’;’SLICY’;’SLICY’;’SLICY’;’SLICY’);
peaks_prob_SLICY = [0.9,0.9,0.9,0.9,0.9,0.9,0.9,0.9,0.9,0.9,0.9,0.9];
peaks_clas_BDRM2 = cellstr(’T62 ’;’BDRM2’;’BDRM2’;’T62 ’;’BDRM2’;’BDRM2’;’T62 ’;’T62 ’;’T62 ’;’T62 ’;’BDRM2’;’BDRM2’;’BDRM2’);
peaks_prob_BDRM2 = [0.85,0.8,0.8,0.8,0.8,0.8,0.8,0.8,0.8,0.8,0.8,0.8];
peaks_clas_BTR60 = cellstr(’BTR60’;’BTR60’;’BTR60’;’BTR60’;’BTR60’;’BTR60’;’BTR60’;’BTR60’;’BTR60’;’BTR60’;’BTR60’;’BTR60’);
peaks_prob_BTR60 = [0.8,0.73,0.8,0.8,0.72,0.73,0.73,0.8,0.8,0.8,0.8,0.8];
peaks_prob_T62 = [0.85,0.8,0.8,0.8,0.8,0.8,0.8,0.8,0.8,0.8,0.8,0.8];
%% Edge Data

```matlab
edge_clas_ZSU = cellstr(['ZSU ';'ZSU ';'ZSU ';'D7 ';'ZSU ';'ZSU ';'... 'ZSU ';'ZSU ';'ZSU ']);
edge_prob_ZSU = [0.72,0.68,0.72,0.8,0.72,0.72,0.72,0.72,0.72,0.72];
edge_clas_ZIL = cellstr(['ZIL ';'ZIL ';'ZIL ';'D7 ';'ZIL ';'ZIL ';'... 'ZIL ';'T62 ';'ZSU ';'ZIL ']);
edge_prob_ZIL = [0.85,0.85,0.8,0.8,0.8,0.8,0.8,0.8,0.8,0.85];
edge_clas_D7 = cellstr(['D7 ';'D7 ';'D7 ';'D7 ';'D7 ';'D7 ';'... 'D7 ';'D7 ';'D7 ']);
edge_prob_D7 = [0.8,0.8,0.8,0.8,0.8,0.8,0.8,0.35,0.8,0.72];
edge_clas_2S1 = cellstr(['2S1 ';'2S1 ';'2S1 ';'2S1 ';'BTR60 ';'2S1 ';'... '2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';'2S1 ';... '2S1 ';'BTR60 '];
e
```
File: sens_array_comb.m

function [sens_clas_comb sens_prob_comb] = 
sens_array_comb(feature_1,prob_1,feature_2,prob_2,feature_3,prob_3)

%%% Information Pertaining to Function
%%% The file is a function used as part of the implementation of
%%% phase one of Dempster-Shafer's Theory of Evidence.
%%% wrap_around.m is the wrap around file. All necessary function calls
%%% are made from that m file.

% Created by Scott Seims

%%
%%
%% PURPOSE:
%% ---The idea was to place all the data for a classification attempt in
%% adjoining elements of the arrays. This is done for both the basic
%% probability assignments and classifications.

%%
%% NOTE:
%% ---This function only works for a system of three sensors.
%%
%%
%% INPUTS:
%% ---The inputs to this function are the arrays coded in data.m.
%% Each call of this function is specific to a target. The feature
%% array is the classification cell array and the prob array is the basic
%% probability assignment array.

%%
%% OUTPUTS:
%% ---This function returns an array where all the data for a classification
%% attempt is placed in adjoining elements of the arrays.

%%

%%

for u=1:10
    k=3*(u-1)+1;
    sens_clas_comb(k)   = feature_1(u);
    sens_clas_comb(k+1) = feature_2(u);
    sens_clas_comb(k+2) = feature_3(u);
    sens_prob_comb(k)   = prob_1(u);
    sens_prob_comb(k+1) = prob_2(u);
    sens_prob_comb(k+2) = prob_3(u);
end

end
% Function: agree.m

function [dem_clas dem_prob] = agree(sens_class, sens_prob)

%% Information Pertaining to Function
% The file is a function used as part of the implementation of
% phase one of Dempster-Shafer's Theory of Evidence.
% wrap_around.m is the wrap around file. All necessary function calls
% are made from that m file.
% Created by Scott Seims
%

% PURPOSE:
% --- In order to calculate the summation of the intersects and null sets it
% is necessary to know which sensors agree. This algorithm figures out the
% agreements. Based off of the agreement specific functions are called in
% order to calculate those two values.
%
% NOTE:
% --- Once again this could is dependent on there existing three sensors in
% the system.
%
% INPUTS:
% --- The arrays created by sens_array_comb are the inputs. This algorithm is
% implemented once per target.
%
% OUTPUTS:
% --- The output of this function is the data fused belief and the
% corresponding classification for all classification attempts specific
% to a target.
%
% dem_prob = zeros(1,10);

for o=1:10
    l=(3*(o-1))+1;

    switch char(sens_class(l))
        case char(sens_class(l+1))
            switch char(sens_class(l+1))
                case char(sens_class(l+2))
                    % fprintf('All agree')
                    sens_class(l:(l+2));
                    [dem_clas(o) dem_prob(o)] = all_agr(sens_class(l:(l+2)), sens_prob(l:(l+2)));
                    % dem_prob_all
                    % dem_clas_all
                    otherwise
                        % fprintf('Two Agree')
                        sens_class(l:(l+2));
                        [dem_clas(o) dem_prob(o)] = two_agr_1_2(sens_class(l:(l+2)), sens_prob(l:(l+2)));
                        % dem_prob_1_2
function [dem_clas dem_prob] = all_agr(sens_class, sens_prob)
    % Information Pertaining to Function
    % The file is a function used as part of the implementation of
    % phase one of Dempster-Shafer's Theory of Evidence.
    % wrap_around.m is the wrap around file. All necessary function calls
    % are made from that m file.
    %
    % Created by Scott Seims
    %
    % GIVEN THAT ALL SENSORS AGREE
    %
    % PURPOSE:
    % ---To calculate the summation of intersections and null sets. Then to use
    % those values to calculate Dempster's Combination Rule.
    %
    % NOTE:
File: two_agr_1_2.m

function [dem_clas dem_prob] = two_agr_1_2(sens_class, sens_prob)

%%% Information Pertaining to Function
%The file is a function used as part of the implementation of
%phase one of Dempster-Shafer's Theory of Evidence.
%wrap_around.m is the wrap around file. All necessary function calls
%are made from that m file.
%
%Created by Scott Seims
%
%GIVEN THAT TWO SENSORS AGREE (SPECIFICALLY 1 WITH 2)
%
%PURPOSE:
%---To calculate the summation of intersections and null sets. Then to use
%  those values to calculate Dempster's Combination Rule.
%
%NOTE:
%---Only can be applied to a three sensor system.
%
%INPUTS:
%---The segment of the both the basic probability assignment array and the
% classification array created by sens_array_comb.m specific to the
% classification attempt undergoing data fusion.
%
%OUTPUTS:
%---The output of this function is the data fused belief and the
% corresponding classification specific to a classification attempt.
%
%% Dempster's Combination Rule
% Mass function of sensor
a1 = sens_prob(1);
a2 = sens_prob(2);
a3 = sens_prob(3);
% Corresponding Unknown mass function of sensor
n1 = 1-a1;
n2 = 1-a2;
n3 = 1-a3;
intersect_1 = a1*a2*n3+a1*n2*n3+n1*a2*n3+n1*n2*n3;
k_1 = 1-a1*a2*a3;
bpa_1 = intersect_1/k_1;
intersect_2 = n1*n2*a3+n1*n2*n3;
k_2 = 1-a1*a2*a3;
bpa_2 = intersect_2/k_2;
if bpa_1 > bpa_2
    dem_clas = sens_class(1);
    dem_prob = bpa_1;
else
    dem_clas = sens_class(3);
    dem_prob = bpa_2;
end
end

File: two_agr_1_3.m

function [dem_clas dem_prob] = two_agr_1_3(sens_class, sens_prob)

%% Information Pertaining to Function
% The file is a function used as part of the implementation of
% phase one of Dempster-Shafer's Theory of Evidence.
% wrap_around.m is the wrap around file. All necessary function calls
% are made from that m file.
%
% Created by Scott Seims
%
%% GIVEN THAT TWO SENSORS AGREE (SPECIFICALLY 1 WITH 3)
%PURPOSE:
%---To calculate the summation of intersections and null sets. Then to use
% those values to calculate Dempster's Combination Rule.
%
%NOTE:
%---Only can be applied to a three sensor system.
%
%INPUTS:
%---The segment of the both the basic probability assignment array and the
% classification array created by sens_array_comb.m specific to the
% classification attempt undergoing data fusion.
%
%OUTPUTS:
%---The output of this function is the data fused belief and the
% corresponding classification specific to a classification attempt.
%

%% Dempster's Combination Rule

%Mass function of sensor
a1 = sens_prob(1);
a2 = sens_prob(2);
a3 = sens_prob(3);

%Corresponding Unknown mass function of sensor
n1 = 1-a1;
n2 = 1-a2;
n3 = 1-a3;

intersect_1 = a1*n2*a3+a1*n2*n3+n1*n2*a3+n1*n2*n3;
k_1 = 1-a1*a2*a3;
bpa_1 = intersect_1/k_1;

intersect_2 = n1*n3*a2+n1*n2*n3;
k_2 = 1-a1*a2*a3;
bpa_2 = intersect_2/k_2;

if bpa_1 > bpa_2
    dem_clas = sens_class(1);
    dem_prob = bpa_1;
else
    dem_clas = sens_class(2);
    dem_prob = bpa_2;
end

end

File: two_agr_2_3.m
function [dem_clas dem_prob] = two_agr_2_3(sens_class, sens_prob)

%% Information Pertaining to Function
% The file is a function used as part of the implementation of
% phase one of Dempster-Shafer's Theory of Evidence.
% wrap_around.m is the wrap around file. All necessary function calls
% are made from that m file.
% %
% % Created by Scott Seims
% %
% % GIVEN THAT TWO SENSORS AGREE (SPECIFICALLY 2 WITH 3)
% %
% PURPOSE:
% %---To calculate the summation of intersections and null sets. Then to use
% % those values to calculate Dempster's Combination Rule.
% %
% NOTE:
% %---Only can be applied to a three sensor system.
% %
% INPUTS:
% %---The segment of the both the basic probability assignment array and the
% % classification array created by sens_array_comb.m specific to the
% % classification attempt undergoing data fusion.
% %
% OUTPUTS:
% %---The output of this function is the data fused belief and the
% % corresponding classification specific to a classification attempt.
% %

%% Dempster's Combination Rule
% Mass function of sensor
a1 = sens_prob(1);    
a2 = sens_prob(2);    
a3 = sens_prob(3);    
% Corresponding Unknown mass function of sensor
n1 = 1-a1;            
n2 = 1-a2;            
n3 = 1-a3;            

intersect_1 = n1*a2*a3+n1*a2*n3+n1*n2*a3+n1*n2*n3;    
k_1 = 1-a1*a2*a3;    
bpa_1 = intersect_1/k_1;    

intersect_2 = a1*n2*n3+n1*n2*a3;    
k_2 = 1-a1*a2*a3;    
bpa_2 = intersect_2/k_2;    

if bpa_1 > bpa_2
    dem_clas = sens_class(2);    
    dem_prob = bpa_1;    
else
    dem_clas = sens_class(1);    
    dem_prob = bpa_2;    
end
function [dem_clas dem_prob] = none_agr(sens_class, sens_prob)

%% Information Pertaining to Function
% The file is a function used as part of the implementation of
% phase one of Dempster-Shafer's Theory of Evidence.
% wrap_around.m is the wrap around file. All necessary function calls
% are made from that m file.
%
% Created by Scott Seims
%
% GIVEN THAT NONE OF THE SENSORS AGREE
%
% PURPOSE:
% ---To calculate the summation of intersections and null sets. Then to use
% those values to calculate Dempster's Combination Rule.
%
% NOTE:
% ---Only can be applied to a three sensor system.
%
% INPUTS:
% ---The segment of the both the basic probability assignment array and the
% classification array created by sens_array_comb.m specific to the
% classification attempt undergoing data fusion.
%
% OUTPUTS:
% ---The output of this function is the data fused belief and the
% corresponding classification specific to a classification attempt.
%
%
%% Dempster's Combination Rule
% Mass function of sensor
a1 = sens_prob(1);
a2 = sens_prob(2);
a3 = sens_prob(3);
% Corresponding Unknown mass function of sensor
n1 = 1-a1;
n2 = 1-a2;
n3 = 1-a3;
intersect_1 = a1*n2*n3+n1*n2*n3;
k_1 = 1-a1*a2*a3;
bpa_1 = intersect_1/k_1;
intersect_2 = n1*a2*n3+n1*n2*n3;
k_2 = 1-a1*a2*a3;
Phase Two and Three:

File: data_creation.m

function data_creation

%% Information Pertaining to Function
% The file is a function used as part of the implementation of phase two and
% phase three of Dempster-Shafer's Theory of Evidence.
% This function is the wrap around file. All necessary function calls
% are made from this m file.
% % Created by Scott Seims
% % % PURPOSE:
% %---The original purpose of this file was to verify the data collection
% % algorithm. This was done using the code commented out below. The output
% % was compared to the phase one data file. The current purpose of this
% % function is to operate as the wrap around. All functions are called
% % from this file. As such this is the only function that needs to be run
% % all other functions just need to be in the directory.
% % % NOTE:
% %---Interesting enough the data verification found that there was an error.
% % However, the error was in the data file used for phase one.
% % % INPUTS:
% %---There are no inputs specific to the wrap around. Initial inputs are
% % added to the workspace through the UserInterface function.
% % % OUTPUTS:
% %---Data fused classification based on target. Ready to be placed into a
% % confusion matrix.
%
end
%% Wrap Around

[headers test sens_num target_num hist_accuracy] = UserInterface;

col_1 = wo_var_data_org(1, headers, test, sens_num, target_num);
col_2 = wo_var_data_org(5, headers, test, sens_num, target_num);

[index_match_col_1] = gen_test(col_1, sens_num, target_num, headers,4);
[index_match_col_2] = gen_test(col_2, sens_num, target_num, headers,8);

[dem_clas_col_1] = dem_numer(1, target_num, sens_num,col_1, col_2, index_match_col_1, test, headers,hist_accuracy);
dem_clas_col_2 = dem_numer(2, target_num, sens_num,col_1, col_2, index_match_col_2, test, headers,hist_accuracy);

%% Verification of data collection working
%Column 6 is prob
%Column 8 is clas
% FstHalf_sens_clas = cell(1,round(target_num/2));
% FstHalf_sens_prob = cell(1,round(target_num/2));
% SndHalf_sens_clas = cell(1,round(target_num/2));
% SndHalf_sens_prob = cell(1,round(target_num/2));
%
% FstHalf_dem_clas = cell(1,round(target_num/2));
% FstHalf_dem_prob = cell(1,round(target_num/2));
% SndHalf_dem_clas = cell(1,round(target_num/2));
% SndHalf_dem_prob = cell(1,round(target_num/2));
%
% for i = 1:round(target_num/2)
%     [FstHalf_sens_clas{1,i} FstHalf_sens_prob{1,i}] =
%     sens_array_comb(headers(1,[1 i];1):col_1[1 i](1):col_1[1 i](2),4,...
%     test(1,[1 i](1):col_1[1 i](2),2),headers(3,[1 i](1):col_1[1 i](2),4,...
%     test(3,[1 i](1):col_1[1 i](2),2),headers(2,[1 i](1):col_1[1 i](2),4,...
%     test(2,[1 i](1):col_1[1 i](2),2));
%     [SndHalf_sens_clas{1,i} SndHalf_sens_prob{1,i}] =
%     sens_array_comb(headers(1,[1 i];1):col_1[1 i](1):col_1[1 i](2),8,...
%     test(1,[1 i](1):col_1[1 i](2),6),headers(3,[1 i](1):col_1[1 i](2),8,...
%     test(3,[1 i](1):col_1[1 i](2),6),headers(2,[1 i](1):col_1[1 i](2),8,...
%     test(2,[1 i](1):col_1[1 i](2),6));
% end
%
% for i =1:round(target_num/2)
%     [FstHalf_dem_clas{1,i} FstHalf_dem_prob{1,i}] = agree(FstHalf_sens_clas{1,i},
%     FstHalf_sens_prob{1,i});
%     [SndHalf_dem_clas{1,i} SndHalf_dem_prob{1,i}] = agree(SndHalf_sens_clas{1,i},
%     SndHalf_sens_prob{1,i});
% end
%
% for i = 1:round(target_num/2)
%     table(:,i) = FstHalf_dem_clas{1,i};
%     table(:,i+4) = SndHalf_dem_clas{1,i};
% end
%
function [headers, test, sens_num, target_num, hist_accuracy] = UserInterface

%% Information Pertaining to Function
% The file is a function used as part of the implementation of phase two and
% phase three of Dempster-Shafer's Theory of Evidence.
% data_creation.m is the wrap around file. All necessary function calls
% are made from that m file.
%
% Created by Scott Seims

% User Interface for the Combination of Algorithms

% PURPOSE:
% --- Inquires into the number of sensors
% --- Inquires into the number of targets
% --- Inquires into the location of necessary data
% --- Allows for user to select data set, data set must match the format as
%   outlined in the readme.
% --- Inquires into the historical accuracy of the identified targets
% NOTE:
% --- Only select the data corresponding to a single sensor at a time
% INPUTS:
% --- User inputs
%   Number of Sensors
%   Number of Targets
%   File Name Containing Data
%   Historical Accuracy
% --- Using the command prompt
% OUTPUTS:
% --- This function outputs all values entered by the user.
% --- The data selected is output as two arrays
%   headers = cell array containing all text data
%   test    = double array containing all numerical data

%% User Interface for Data Retrieval

sens_num = input('Please enter the number of sensors in the system.
');
target_num = input('Please enter the number of targets in the system.
');

hist_accuracy = zeros(1, sens_num);

for i=1:sens_num
    filename = input('Please type in the name of the file containing data for each sensor
');
    [test(i,:), headers(i,:)] = xlsread(filename,-1);
    hist_accuracy(1,i) = input('Please type in the corresponding historical accuracy of the above
');
end

%
function [target] = wo_var_data_org(column, headers, test, sens_num, target_num)

%% Information Pertaining to Function
% The file is a function used as part of the implementation of phase two and
% phase three of Dempster-Shafer's Theory of Evidence.
% data_creation.m is the wrap around file. All necessary function calls
% are made from that m file.
%
% Created by Scott Seims
%
% PURPOSE:
% ---This function determines the placement of the data specific to each
% target. This is done by column. By coding it based on column number
% it is possible to have as many columns as necessary within the data
% detailing the sensors.
%
% NOTE:
% ---The wrap around is designed for at
% most two columns at the moment.
%
% INPUTS:
% ---The first column of the target data column
% Column = 1 for the first target data column
% Column = 5 for the second target data column
% --- Both the headers and test arrays from UserInterface.m
% ---The number of sensors and targets within the system
% OUTPUTS:
% ---The output is an array containing the start and stop indices for each
% target in the identified target data column.
%

%% Target Data Placement Algorithm

% The assumption is the data will be split into two columns and the number of
% targets may not be even. If the number of targets entered into the
% algorithm is double the actual amount of data placed in only one column can
% be used with no change to the code.
col_split = round(target_num/2);

% Declaration of the Cell Array
target = cell(sens_num,col_split);

for k = 1:sens_num
    % Searching for the empty indexes placed between each target to identify
    % the end of the information pertaining to a single target
    D = cellfun('isempty',headers(k,:,4));
    count = 1;
    for i = 1:size(headers,2)
```matlab
if i == 1
    start = i;
end
if D(i)== 1
    j = i + 1;
    target{k, count} = [start, i-1];
    count = count + 1;
    start = j;
end
if i == size(headers,2)
    target{k, count} = [start, i];
end
end
end

File: gen_test.m

function [index_match] = gen_test(col, sens_num, target_num, headers,agree_var_col)

    %% Information Pertaining to Function
    %The file is a function used as part of the implementation of phase two and
    %phase three of Dempster-Shafer's Theory of Evidence.
    %data_creation.m is the wrap around file. All necessary function calls
    %are made from that m file.
    %
    %Created by Scott Seims
    %
    %PURPOSE:
    %---This is an algorithm thaat is used to determine and
    % record which sensors agree and which disagree for the calculation of
    % the intersect and null sets. To keep the processing
    % speed relatively high and the space requirement low, the algorithm compares
    % the classification of each sensor and recods the index of the classifier
    % as related to the depth within the headers array.
    %
    %NOTE:
    %---If for any reason a change is made to the code it is much easier to
    % debug if run using a single target. The orientation of the array is
    % counterintuitive. Good luck.
    %---For implementation in regards to the first target data column
    % agree_val_col = 4
    %---For implementation in regards to the second target data column
    % agree_val_col = 8
    %INPUTS:
    %---The inputs into this function are the number of sensors, the number of
    % targets, the ouput of wo_var_data_org.m, and the column containing the
    % classification data
    %
    %OUTPUTS:
    %---The output of this function is an array that contains the indices of
    % the sensors that agree for each potential target focus.
    %
    ```
Agreement Set Identification

Remember to set j, which is stepping through the index tracking
Remember to set k, which is adding the appropriate number to the occurrence

agreement_set = cell(1,sens_num);
index_match = cell(round(target_num/2),range(col{1,1})+1,sens_num, sens_num);

for j = 1:round(target_num/2)
    for k=0:range(col{1,j})
        for i = 1:sens_num
            agreement_set{1,i} = headers(i,[col{1,j}(1)+k],agree_var_col); %removed the j term
        end
    end
end

p is equal to the number of different target assignments
k is the image pass/target pass depending how you look at it
h corresponds to the number of sensors that agree

early_agree = 'no';
p=0;
o_before=1;
h=1;
count =1;
for o =1:sens_num
    t=1;
    compare = agreement_set{1,o};
    switch char(agreement_set{1,o})
        case early_agree
        otherwise
            h=1;
            p = p+1;
            index_match{j,k+1,p,h} = o;
            agreement_set{1,o} = 'no';
        end
    end
while t <= sens_num
    if t==o & t < sens_num
        t=t+1;
    end
    if t~=o
        switch char(compare)
            case early_agree
            case char(agreement_set{1,t})
                h = h+1;
                agreement_set{1,t} = 'no';
                index_match{j,k+1,p,h} = t;
            otherwise
                end
        end
    end
end
function [dem_clas] = dem_numer(col, target_num, sens_num, col_1, col_2, index_match, test, headers, hist_accuracy)

%% Information Pertaining to Function
% The file is a function used as part of the implementation of phase two and
% phase three of Dempster-Shafer's Theory of Evidence.
% data_creation.m is the wrap around file. All necessary function calls
% are made from that m file.
% Created by Scott Seims

% PURPOSE:
% ---This function creates the arrays of the probabilities or beliefs of the sensors
% that agree using the output of gen_test.m. This is done for each
% classification attempt of each target.

% NOTE:
% ---As with all the other algorithms this is done on the per column bases.
% It is necessary to call this function for every target data column

% INPUTS:
% ---col identifies the target data column. Again this function is passed
% the number of sensors and targets, the historical accuracy, the headers
% array and the test array. It is passed the output of
% wo_var_data_org.m for both target data columns. It is also passed the output of
% gen_test.m.

% OUTPUTS:
% ---The output is the data fused classification for each classification
% attempt of each target

% Probability Agreement Set Creation

%------------------------------------------------------------%-% T=1;
% image = 2;
% col = 1;
% index_match = index_match_col_1;
%------------------------------------------------------------%
%for testing col with be set to col_1
if col == 1
    col = col_1;
    var_col = 2;
    clas_col = 4;
else
    col = col_2;
    var_col = 6;
    clas_col = 8;
end

for T=1:round(target_num/2)
    for image=1:range(col{1,T})+1
        for agree_set=1:sens_num
            for L=1:sens_num
                potential = index_match(T,image,agree_set,:);
                if isempty(potential{L}) ~= 1
                    prob(agree_set, L) = test(potential{L},(col{1,T}(1)-1+image),var_col);
                    *hist_accuracy(potential{L});
                    prob_neg(agree_set, L) = (1 - test(potential{L},(col{1,T}(1)-1+image),var_col));
                    *hist_accuracy(potential{L});
                end
            end
        end
    end
end

%This will be the dem_prob
%Depending on the function called here the implementation is either
%phase two or three. Phase two uses real_t_table, while phase three
%uses real_t_table_weighted.
[true_num] = real_t_table(prob,sens_num);

prob = zeros(agree_set,L);
prob_neg = zeros(agree_set,L);

true_index = index_match(T,image,true_num(2),1);
true_num = [0 0];
%Classification saved
dem_clas(T,image) = headers(true_index{1},(col{1,T}(1)-1+image),clas_col);
end
end

File: real_t_table.m

function [true_num] = real_t_table(prob,sens_num)

%%% Information Pertaining to Function
The file is a function used as part of the implementation of phase two of Dempster-Shafer's Theory of Evidence. data_creation.m is the wrap around file. All necessary function calls are made from that m file. 

Created by Scott Seims

PURPOSE:
--- When calculating the summation of the intersects for Dempster's Combination Rule it is necessary to determine the permutations for each agreement set. This is done using a truth table.

NOTE:
--- This file is used only if called from dem_numer. Check there if not sure which of the two truth table algorithms are being called

INPUTS:
--- The input to this file is the number of sensors in the system and the array containing the belief values for all agreement sets as determined by gen_test.m

OUTPUTS:
--- The output of this algorithm is the data fused belief and the sensor index that has the corresponding classification.

Permutations Algorithm (Summation of the Intersections)
true_size = zeros(1, size(prob,1));
pre_mult_table = zeros(1,1);
um = zeros(length(true_size),1);
for dem_depth = 1:size(prob,1)
  for dem_length = 1:size(prob,2)
    if prob(dem_depth, dem_length) ~= 0
      true_size(1,dem_depth) = true_size(1,dem_depth) + 1;
    end
  end
end
for test_dem_prob = 1:length(true_size)
  test_dem_prob = 1; %this is for TEST
  sizer = true_size(test_dem_prob);
  flop =0;
  t_table = zeros((2^(sizer-1))*2,sizer);
  for j = 1:sizer
    modify = 1;
    val = prob(test_dem_prob,j);
    agree = prob(test_dem_prob,j);
    not_agree = 1-prob(test_dem_prob,j);
    flop = 2^(sizer - j);
    for i = 1:(2^(sizer-1))*2
      if modify == 0
        if val == agree
          
        
      
    end
  end
end
val = not_agree;
else
    val = agree;
end
end
t_table(i,j) = val;
modify = rem(i,flop);
end
end
t_table_depth = (2^(sizer-1))*2;
for concat = 1:length(true_size)
    for concat_length = 1:true_size(concat)
        if concat ~= test_dem_prob
            concatenate = ones(t_table_depth,1)*(1-prob(concat,concat_length));
            pre_mult_table = [t_table,concatenate];
            t_table = pre_mult_table;
        end
    end
end
pre_mult_table = zeros(10,10);
pre_num = ones(t_table_depth,1);
for mult_depth=1:t_table_depth
    for mult_length=1:sens_num
        pre_num(mult_depth,1) = pre_num(mult_depth,1)*t_table(mult_depth,mult_length);
    end
end
for sum_depth = 1:t_table_depth-1
    num(test_dem_prob) = num(test_dem_prob) + pre_num(sum_depth);
end
end
% num = num./
true_num = [num(1) 1];
for true_dem_prob_num = 1:length(true_size)
    if num(1) < num(true_dem_prob_num)
        true_num = [num(true_dem_prob_num) true_dem_prob_num];
    end
end
% true_num = [num(1) 1];
%   % for true_dem_prob_num = 2:length(true_size)
%       if num(true_dem_prob_num) > num(true_dem_prob_num-1)
%           true_num = [num(true_dem_prob_num) true_dem_prob_num];
%       end
%   % end
% end
function [pot_true_num] = real_t_table_weighted(prob,sens_num,prob_neg,target_num)

%% Information Pertaining to Function
%The file is a function used as part of the implementation of
%phase three of Dempster-Shafer's Theory of Evidence.
data_creation.m is the wrap around file. All necessary function calls
%are made from that m file.
%%Created by Scott Seims
%

%% PURPOSE:
%---When calculating the summation of the intersects for Inagaki's Unified
%   Combination Rule it is necessary to determine the permutations for each
%   agreement set. This is done using a truth table.

%% NOTE:
%---This file is used only if called from dem_numer. Check there if not
% sure which of the two truth table algorithms are being called

%% INPUTS:
%---The input to this file is the number of sensors in the system and the
% two arrays containing the belief values for all agreement sets as determined by
% gen_test.m. prob contains the beliefs that the target is as classified
% while prob_neg contains the compliment. Both are scaled by the
% historical accuracy.

%% OUTPUTS:
%---The output of this algorithm is the data fused belief and the sensor
% index that has the corresponding classification.

%% Permutations Algorithm (Summation of the Intersections)
true_size = zeros(1, size(prob,1));
pre_mult_table = zeros(1,1);
num = zeros(length(true_size),1);
num_combRule = zeros(length(true_size),1);

for dem_depth = 1:size(prob,1)
    for dem_length = 1:size(prob,2)
        if prob(dem_depth, dem_length) ~= 0
            true_size(1,dem_depth) = true_size(1,dem_depth) + 1;
        end
    end
end

q_null_1 = 1;
q_null_2 = 1;
q_null = 1;
%Calculating the null value
for null_calc = 1:length(true_size)


97
if true_size(null_calc) == sens_num
    q_null = 0;
else
    for step_null = 1: true_size(null_calc)
        q_null_1 = q_null_1 * prob(null_calc, step_null);
        q_null_2 = q_null_2 * prob_neg(null_calc, step_null);
    end
end
if q_null == 1;
    q_null = q_null_1 + q_null_2;
end
for test_dem_prob = 1: length(true_size)
    %test_dem_prob = 2; % this is for TEST
    %---------------------------------------------
    flop = 0;
    % Creation of the appropriate agreement set arrays based on the three
    % states rather than the two used in the other truth table algorithm
    for concat = 1: length(true_size)
        if true_size(concat) == sens_num
            prob_complete = prob(test_dem_prob, [1: true_size(test_dem_prob)]);
            prob_neg_complete = prob_neg(test_dem_prob, [1: true_size(test_dem_prob)]);
            prob_neg_complete = prob_neg_complete / target_num;
        end
        if concat ~= test_dem_prob
            concatenate_neg = zeros(1, true_size(concat));
            concatenate_to_neg = zeros(1, true_size(concat));
            for concat_length = 1: true_size(concat)
                concatenate_neg(1, concat_length) = prob_neg(concat, concat_length);
                concatenate_to_neg(1, concat_length) = prob(concat, concat_length);
                prob_complete = [prob(test_dem_prob, [1: true_size(test_dem_prob)]), concatenate_neg];
                prob_neg_complete = [prob_neg(test_dem_prob, [1: true_size(test_dem_prob)]), concatenate_to_neg];
                prob_neg_complete = prob_neg_complete / target_num;
            end
        end
    end
    % Creation of truth table
    sizer = length(prob_complete);
    t_table = zeros((2^sizer - 1) * 2, sizer);
    for j = 1: sizer
        modify = 1;
        val = prob_complete(j);
        agree = prob_complete(j);
        not_agree = 1 - (prob_complete(j) + prob_neg_complete(j));
        flop = 2^sizer - j;
        for i = 1: (2^sizer - 1)
            if modify == 0
                if val == agree
                    val = not_agree;
                else
                    val = agree;
                end
            else
                val = not_agree;
            end
            t_table(i, j) = val;
        end
    end
val = agree;
end
end
t_table(i,j) = val;
modify = rem(i,flop);
end
end
t_table_depth = (2^(sizer-1))*2;
pre_mult_table = zeros(1,1);
pre_num = ones(t_table_depth,1);
q_X = 1;

for mult_depth=1:t_table_depth
  if mult_depth == t_table_depth
    for mult_length=1:sizer
      q_X = q_X*t_table(mult_depth,mult_length);
    end
  end
  if mult_depth ~= t_table_depth
    for mult_length=1:sizer
      pre_num(mult_depth,1) = pre_num(mult_depth,1)*t_table(mult_depth,mult_length);
    end
  end
end

for sum_depth = 1:t_table_depth-1
  num(test_dem_prob) = num(test_dem_prob) + pre_num(sum_depth);
end

comp_pre_con = 10000000000000000000000000;
pre_conflict = (1/(1-q_null-q_X));
if pre_conflict < comp_pre_con
  comp_pre_con = pre_conflict;
end

end

pot_true_num = [0 0];
true_num = [0 0];

%Calculation of Inagaki's Unified Combination Rule
for conflict_parameter = 0:(comp_pre_con/1000):comp_pre_con
  for step = 1:length(true_size)
    num_combRule(step) = (1 + conflict_parameter*q_null)*num(step);
  end
end

if true_num(1) > pot_true_num(1)
  pot_true_num = true_num;
end
true_num = [num_combRule(1) 1];
for true_dem_prob_num = 1:length(true_size)
  if true_num(1) < num_combRule(true_dem_prob_num)
    true_num = [num_combRule(true_dem_prob_num) true_dem_prob_num];
  end
end
end
end
end

% true_num = [num(1) 1];
%
% for true_dem_prob_num = 2:length(true_size)
%   if num(true_dem_prob_num) > num(true_dem_prob_num-1)
%       true_num = [num(true_dem_prob_num) true_dem_prob_num];
%   end
% end