Dismount Threat Recognition Through Automatic Pose Identification

THESIS

Andrew M. Freeman, Captain, USAF

AFIT/GE/ENG/12-14

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GE/ENG/12-14

Dismount Threat Recognition Through Automatic Pose Identification

THESIS

Presented to the Faculty
Department of Electrical and Computer Engineering
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Electrical Engineering

Andrew M. Freeman, Master of Business Administration
Captain, USAF

March 2012

AFIT/GE/ENG/12-14

Dismount Threat Recognition Through Automatic Pose Identification

Andrew M. Freeman, Master of Business Administration
Captain, USAF

Approved:

/signed/	March 2012
Lt.Col. Jeffrey D. Clark, PhD (Chairman)	Date
/signed/	March 2012
Dr. Angela A. Sodemann (Member)	Date
/signed/	March 2012
Dr. Gilbert L. Peterson (Member)	Date

AFIT/GE/ENG/12-14

Abstract

The U.S. military has an increased need to rapidly identify and combat non-conventional adversaries. Dismount detection systems are being developed and expanded to provide more information and identify any potential threats. Current work in this area utilizes multispectral imagery to exploit spectral properties of exposed skin and clothing. These methods are useful in the location and tracking of dismounts but do not directly discern a dismounts level of threat. However, by performing pose recognition and identification it can be possible to discern this threat information. Pose recognition is the process of observing a scene through an imaging device(s), determining that a dismount is present, identifying the three dimensional (3D) position of the dismount's joints, and evaluating what the current configuration of the joints means. This thesis explores the use of automatic pose recognition to identify threatening poses and postures by means of an artificial neural network. The data is collected utilizing the depth camera and joint estimation software of the Kinect for Xbox 360. A threat determination is made based on the pose identified by the network. Accuracy is measured both by the correct identification of the pose presented to the network, and proper threat discernment. The end network achieved approximately 81% accuracy for threat determination and 55% accuracy for pose identification with test sets of 26 unique poses. This disparity is a result of commonly confused poses in the network being of the same threat level as the pose presented. Overall, the high level of threat determination accuracy indicates that automatic pose recognition is a promising means of discerning whether a dismount is threatening or not.

Acknowledgements

I would like to thank my friends and family for their continued support throughout this endeavor. My AFIT peers deserve a lot of thanks for helping to keep me focused on my work and providing entertaining diversion as needed. Thanks also go to my advisor Lt. Col. Jeffrey Clark for keeping me on track with my research and providing much needed direction and commentary.

Andrew M. Freeman

Table of Contents

List of Figures

Figure Page

Figure Page

List of Tables

Dismount Threat Recognition Through Automatic Pose Identification

I. Introduction

The United States Air Force has listed one of its top priorities as "getting more intelligence, surveillance and reconnaissance to the war zone" [6]. This priority is a direct result of encounters with unconventional adversaries, such as individuals (dismounts) dressed in civilian attire attacking crowds of people. The Joint Improvised Explosive Device Defeat Organization (JIEDDO), Air Force Research Laboratories Layered Sensing Technology Division, and other military organizations are interested in the identification of dismounts and the ability to discern their intentions [7]. Dismount detection systems have the potential to provide information pertaining to a dismount's capability to do harm, intended target(s), and opportunities. The information provided by these systems can be used to predict and prevent hostile actions against friendly assets.

Dismount detection and recognition involves identifying the presence of a dismount, discerning physical characteristics and attributes, assessment of the surrounding situation, and analyzing the information. The Sensors Exploitation Research Group at the Air Force Institute of Technology at Wright Patterson Air Force Base, Ohio has created several methods to detect and track dismounts within surveillance data [8–13]. Brooks, Koch, Nunez and Poskosky [8, 11–13] develop and refine skin-cued dismount detection systems through the use of multispectral image analysis. Climer [10] expands the applicability of the skin-cued dismount detection by overcoming certain pose limitations. Clark [9] uses a stochastic non-redundant feature selection method on hyperspectral data sets to detect and characterize clothing. The automobile industry continues to refine pedestrian avoidance systems to assist drivers [14, 15]. Microsoft's Kinect for Xbox 360 [3] uses real-time human pose recognition to provide more interactive gaming experiences where the human body is the controller [5]. All of these efforts provide pieces of information useful to dismount threat recognition.

Much of the work performed at AFIT [8, 10–13] deals primarily with detecting the presence of dismounts within imagery using information collected in the near-infrared (NIR) and short wave infrared (SWIR). Different types of dismount characterization and classification are needed to provide threat information. Aspects such as equipment being carried, posturing and movement patterns, physical attributes of the dismount, and the surrounding situation are indicators that aid the threat recognition process.

1.1 Problem Statement

Many actions that humans perform can be broken down into sub actions or sequences of events. If an observer sees a weapon being drawn, they assume it is likely to be used in the near future and that the situation has become dangerous. The intent of this thesis is to enable a computer system to observe and recognize dangerous situations. According to Karl Rehn, a National Rifle Association (NRA) instructor, and his associates [16] firing a handgun accurately at a target is a seven stage sequence:

1. gripping the gun in its holster,
2. drawing it up out of the holster,
3. pointing the gun directly at the target,
4. bringing your other hand into position forming a solid two-handed grip and clicking off the safety,
5. sighting the gun at the target,
6. pulling the trigger until a shot is fired,
7. and taking your finger off the trigger and flicking the safety back on until ready to fire again.

Recognizing the first stage, when a hand is gripping a weapon and preparing to draw, provides valuable warning time for bystanders and personnel to seek cover or intervene. Gripping a weapon typically involves one of several common postures. These postures can be observed using the method developed by Shotton *et al* [5] and mapped as 3-dimensional (3D) joint positions. Their method was chosen because it produces joint estimates in real time, is readily available using commercial devices and software, and is unaffected by body shape and clothing.

Breaking down hostile situations into a sequence of events reveals indicators and markers that are used to predict and prevent the situation. This thesis focuses on identifying the actions that lead up to dangerous or threatening events and developing a way to detect those actions automatically. The goal is to provide early warning that an event may occur and a confidence level at which the prediction is made. An ontological view of human behavior is used throughout this thesis as a means of determining how different actions fit together in order to comprise the entire situation. Training data is collected using the Kinect sensor [3] and supporting software to map joint positions. Machine learning methods are used for pose identification and recognition.

1.2 Scope

The focus of the research is centered on

- Identifying the behaviors and postures that precede threatening actions/activities;
- Designing and training an algorithm to detect these behaviors and postures; and
- Determining a classification and identification accuracy.

Behavior and posture identification is shown as part of the literature review in Section 2.1.3.3 and Section 2.2.1. The identified postures and behaviors represent a small subset of possibilities and should not be viewed as all inclusive. Potential algorithms and training methods are viewed in Section 2.2.3 and expanded upon in Section 3.3. The method used and reasoning behind the selection can be found in Chapter III. How the classification accuracy is determined is presented in Section 4.1 with the end results shown in Section 4.4.

1.3 Document Organization

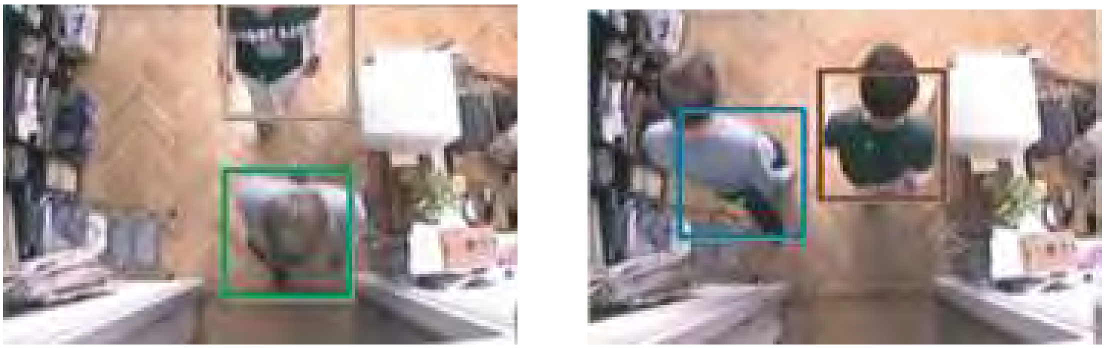
Chapter II provides background information and discusses other efforts of similar nature. Chapter III describes the approach used to solve the problem, assumptions made, and general methodology. Chapter IV presents the results of this thesis and the analysis of what those results mean. Chapter V summarizes the results of this thesis effort and makes recommendations for future work.

II. Background

This chapter provides an overview of previous dismount threat recognition methods and the necessary background for the work in this thesis. Various methods that are used to detect the presence of a dismount within imagery, video, and other surveillance data is discussed. A discussion on the dismount characterization information that these detection methods provide is covered. A closer look at information useful in making a threat level assessment is provided. Finally, information related to the method chosen for development in this thesis and the technologies that enable it is covered.

2.1 Dismount Detection and Characterization Methods

2.1.1 Detection Methods. Dismount detection is achieved in a couple different ways. Detection can be accomplished using motion detection, gateway crossings, skin detection, or other methods not discussed here. Motion detection involves analyzing two or more frames, or images taken over a short duration to determine changes. Stationary cameras work best for motion detection, but cameras with set locations and known rates of movement can be incorporated. Draganjac *et al.* [1] use a form of change detection by placing a camera directly above an entry point imaging the entry area. After removing the background information, the changed pixel clusters are compared to various models to determine if a dismount is present. When a dismount is perceived, they are tracked, within the systems field of view, as shown in Fig. 2.1. Gateway crossing systems are commonly used and employed in many retail stores. For example, a pair of sensors are placed on either side of a doorway or entry point linked by an infrared beam. Anytime the beam is interrupted an alarm is activated, indicating an entry or exit from the store. Gateway systems have the potential to electronically interrogate dismounts as they traverse the designated operating area. However, they have limited applicability since their operating area is limited and they can be evaded or spoofed. Human skin has specific properties when viewed with short wave infrared wavelengths and the visible spectra. Skin detection methods search through frames of data to locate these properties [8, 10–13]. The method used in [1] looks specifically for faces using computer models and feature extraction techniques. The work in [8, 10–13] look for pixels that match skin characteristics in order to narrow the field of view. Then they analyze relative positions and sizes of clusters to confirm the presence of a dismount.

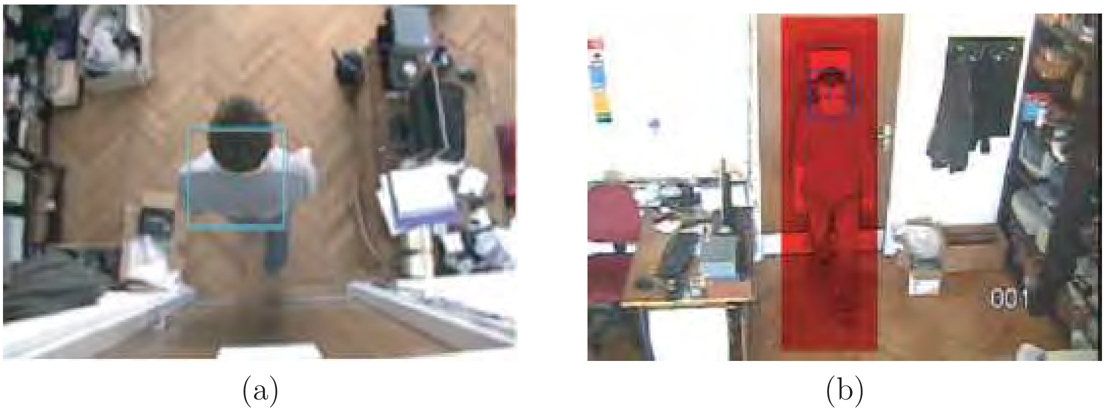


(a) (b)

Figure 2.1: Examples of dismount detection and tracking.

2.1.2 Detection Method Evaluation. Each detection method provides information about the dismounts present in the area. Motion detection and tracking provides pathing data and changes within a scene. The path data may be used to determine whether a dismount has been following a regular path, for example, planning an attack or scoping out an area. The change detection portion highlights introduction of new objects to a scene or the disruption of objects to indicate tampering or even placement of improvised explosive devices. A gateway system can provide the number of dismounts in an area, rate of entry and exit, and electronically interrogate dismounts that pass through. When a hostile event occurs it is useful to know the number of people that are involved in the situation in case someone is hiding or knocked unconscious. High variances in rates of flow may indicate a situation has arisen causing people to flee the scene; or that hostile dismounts have arrived at the scene. Flow stoppages during regular hours of operation may also indicate a hostage situation. Electronic interrogation may discover concealed weapons or explosive devices as Grafulla-Gonzalez *et al.* discuss in [17].

One application of skin detection is to identify malicious intent. In many societies around the world it is not customary to purposely hide your face while participating in day to day activities such as walking around town, visiting with friends, or going to the store. However, if a dismount is evading cameras by purposely covering their face, the amount of exposed skin is reduced and can be detected which may indicate a threat. The system proposed by Draganjac *et al.* [1] uses two cameras. The first camera is used for motion detection to indicate the presence of a dismount. The second camera is used to scan for faces on the dismounts indicated. This method is illustrated in Fig. 2.2. Some key problems with this methodology are that cultural differences and weather variances are not taken into account. In some cultures it is normal practice to cover the face with a veil. Also, not every region has fair weather all the time; therefore, face covering is worn as protection from the elements.



(a) (b)

Figure 2.2: Example of two part system developed by Draganjac *et al.* [1]. The first camera in (a) allows the detection and tracking of a dismount. The second camera in (b) searches for a face to see if it is concealed or not.

Another application of skin detection is identification of race and ethnicity. The primary reasoning behind this type of identification is to determine racial composition of a crowd, and identifying individuals or groups that are not of similar ethnic origins to their surroundings. If ethnic origin is not the same as those around them, it may mean they are trying to infiltrate a specific location. This, by itself, is not a strong indicator of a threat. There are a wide variety of legitimate reasons that dismounts of multiple ethnicities would be in the same location, i.e. tourism and foreign exchange programs, etc.. In the grand scheme of dismount threat recognition (DTR) this application of skin detection plays a minor role and merely helps serve as a contributing factor to an integrated system approach.

2.1.3 Other Dismount Characterization Tools and Methods.

2.1.3.1 Object Detection. The objects dismounts carry play a large role in the actions they are able to take. Some objects such as a gun held openly in a dismounts hand, are clear signs of a threatening capability. However, objects and equipment may not always be so readily visible. Equipment may be concealed under baggy clothing, or carried inside a backpack. For the non-concealed case, a dismount carrying weapons or explosives openly can be considered a threat. This type of threat is easily determined with video surveillance actively monitored by a human. However, it is impossible to guarantee that someone will be actively watching the appropriate monitors at the right time. A system that extrapolates this information from the data and then alerts a human in the loop to verify and respond is more effective.

A more difficult situation is that of concealed objects. For example, a suicide bomber wearing a vest underneath his/her clothing. A method to detect the presence of a hidden vest or presence of explosives would be highly useful. One useful short range tool utilizes properties inherent to all things like the dielectric constant [17]. This tool uses millimeter-wave personnel scanners to determine the electromagnetic properties of people and objects passing through the sensor field of view. These properties are then referenced against tables of known attributes to determine the composition of the objects. Objects composed of materials that are threatening generates an alarm [17]. The primary limitation of the millimeter-wave method is that it requires a gateway system, as mentioned in Section 2.1.1, to be effective. If the dismount evades the gateway or the emplacement of a gateway is simply not feasible, then it cannot be applied.

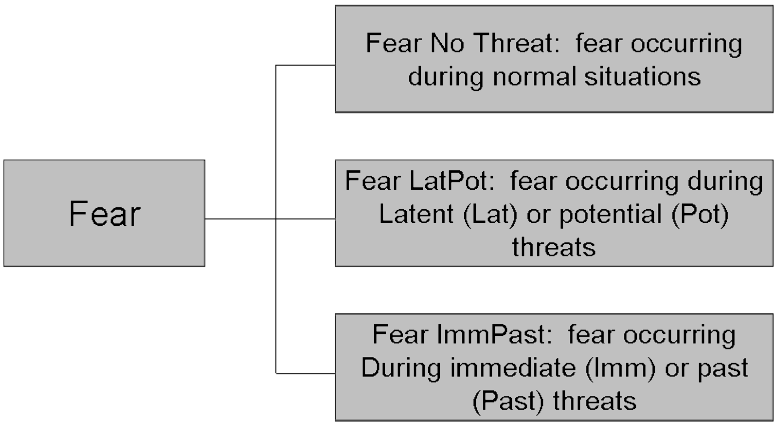

Fear
Fear No Threat: fear occurring during normal situations
Fear LatPot: fear occurring during Latent (Lat) or potential (Pot) threats
Fear ImmPast: fear occurring During immediate (Imm) or past (Past) threats

Figure 2.3: Fear subclasses, as determined by Clavel *et al.* [2], that can be determined through analysis of vocal patterns.

2.1.3.2 Acoustic Modeling. Work has been performed in the modeling of stress and fear as indicated in speech patterns [2]. Often, it is known to a crowd that something is amiss before a tragic event occurs. Monitoring speech patterns of dismounts for these stress and fear indicators is used to indicate a possible event occurring in the near future. The paper by Clavel [2] claims to have an accuracy of 70.3% in determining fear from vocal response. The categorization of the three subclasses of fear is shown in Fig. 2.3. This method requires:

- live audio data from the protected location,
- dismounts aware of, or with the impression that, a dangerous event is going to occur,
- and enough fear to be present in their vocal patterns.

This system may help in small scale situations but may be too limited to be applicable for large scale problems.

2.1.3.3 Human Behavioral Ontology. Human Behavioral Ontology operates on the premise that the actions taken by a target individual or a group give information that implies their intentions and future actions [18]. These action-intention groupings serve as a starting point in true dismount threat recognition. In order to make inferences from data, it is necessary to understand what the data means [18]. Cohen [18] seeks to model data retrieved through visual imagery as human motion behaviors. Determining a subset of minor gestures, which meld into motions and actions of an individual, an estimation of the intentions of the dismount can be made [18]. Taking these motions to the next stage, Cohen's procedure is applied to individuals and crowds. The motions and actions are classified as normal or aberrant, strange and possibly threatening, based on models. The threatening posture of these motions are determined when compared to data collected preceding past hostile actions [19]. Through the use of learning algorithms and event analysis of both simulation and actual events, systems can learn to more accurately analyze and predict certain situations [19]. One system uses a primary threat detection ontology (TDO) with a secondary threat detection learning ontology (TDLO) which accomplishes an analysis to provide more accurate information for the TDO [20]. While these methods are very powerful, it does take a large effort to gather the needed data and teach these ontology based protection systems to appropriately identify threats.

2.2 Pose Estimation and Enabling Technologies

2.2.1 Pose Estimation Related Work. Dismount pose estimation has been a topic of interest for security, human-computer interaction, gaming, computer animation and simulation, and more for several years. Recent advances in this field are due to the advent of real-time depth cameras. Grest *et al.* [21] use depth cameras to track a skeleton of a specific size and known starting position. Anguelov *et al.* [22] use 3D range scan data to segment puppets into head, limbs, torso, and background. Zhu & Fujimura [23] build detectors using heuristics for coarse upper body estimation of head, torso, and arms but require a T-pose to initialize the model. Plagemann *et al.* [24] build a 3D mesh to find extrema interest points which are then classified as head, hand, or foot. Shotton *et al.* [5] predict 3D joint positions using a single depth image through an object recognition approach, and the design of intermediate body part representations. The approach developed in [5] forms the basis for data collection efforts in this thesis. Fig. 2.4 is a graphical representation of the data output by

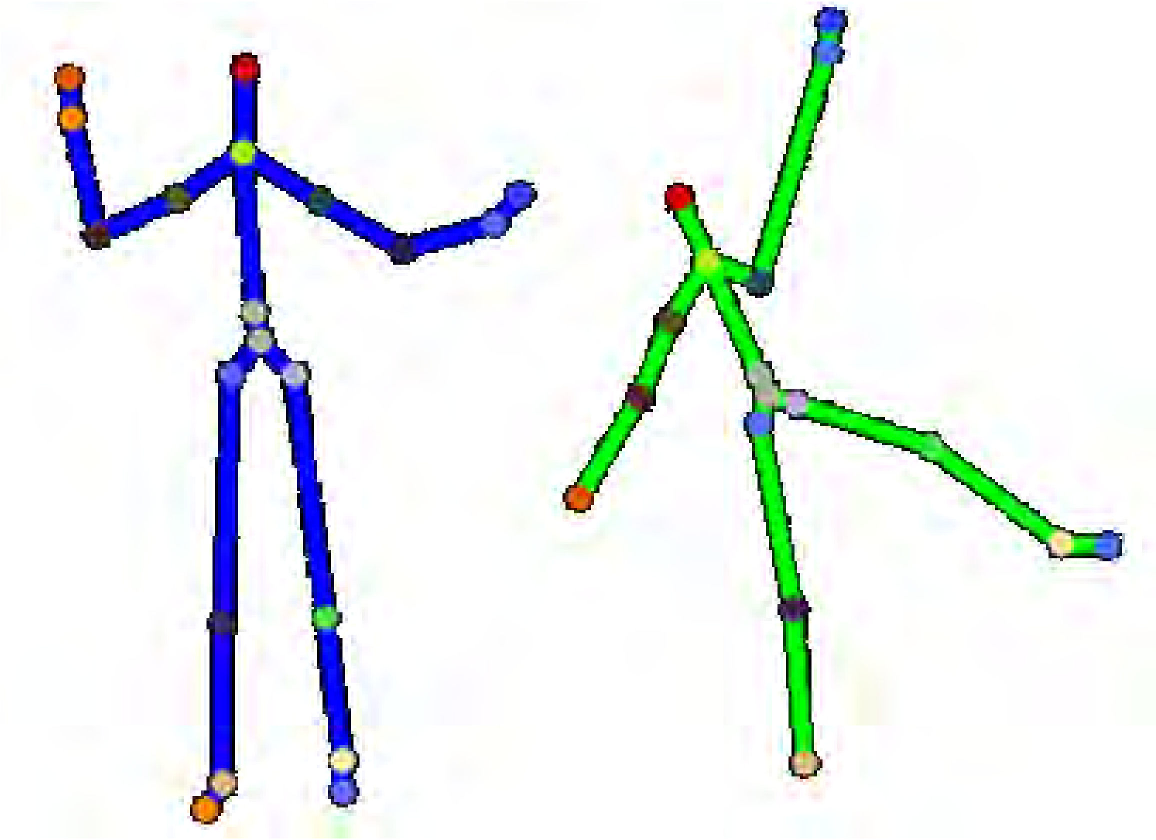


Figure 2.4: Example joint estimates and skeleton mapping produced by the Kinect sensor [3].

the Kinect sensor [3]. The sensor outputs 3D joint position estimates for: feet, ankles, knees, hips, hands, wrists, elbows, shoulders, head, neck, waist, and torso. The methods described all strive to identify the position of the human body within the scene, none of them associate a meaning or intent with the poses. The novel aspect of this research is the interpretation of these poses and creating a link between what the computer sees and the level of threat implied.

2.2.2 Enabling Technologies. Early efforts in pose estimation were hampered by the difficulty in creating realistic intensity images through computer graphics, due to the color and texture variability present from clothing, hair, and skin. It was often necessary to resort purely to 2D silhouettes. Joint estimates using motion capture equipment are useful for animation efforts but impractical for field use due to the requirements of sensors being placed on the subject of interest, as shown in Fig. 2.5. The introduction of depth cameras removes the need for intermediate sensors or 3D extraction from 2D information. Depth cameras use various methods to discern the distance of each pixel necessary for 3D. Kinect [3] uses an infrared laser projector and a monochrome CMOS sensor producing a 640x480 image at 30 frames per second with a depth resolution of a few centimeters [5]. Depth cameras offer several advantages over intensity sensors:

- they work in low light levels,
- give a calibrated scale estimate,
- are color and texture invariant,
- resolve pose ambiguities from silhouettes,
- are easy to synthesize,
- and simplify background subtraction [5].

Samples of real and synthetic data, produced by Shotton *et al.* [5] is shown in Fig. 2.6. Notice the variety of body shapes, poses, clothing, and amount of cropping present in the images.

2.2.3 Associative Memory Neural Networks. Memories are formed from experiences. An event happens and the human brain makes changes so that the event is remembered. An associative memory neural network (AMNN) seeks to mimic this process using computers [25]. AMNNs are trained by passing through several input vectors with predefined outcomes. These inputs are correlated with their respective outputs through multiplication and then summed across the range of inputs to create a memory matrix. Memories in the human brain can get mixed up and slightly confused with each other, in an AMNN this is called crosstalk. Crosstalk occurs when the weight matrix associated with one input and output pair contains information, that when summed together, corrupts or alters another inputs respective values.

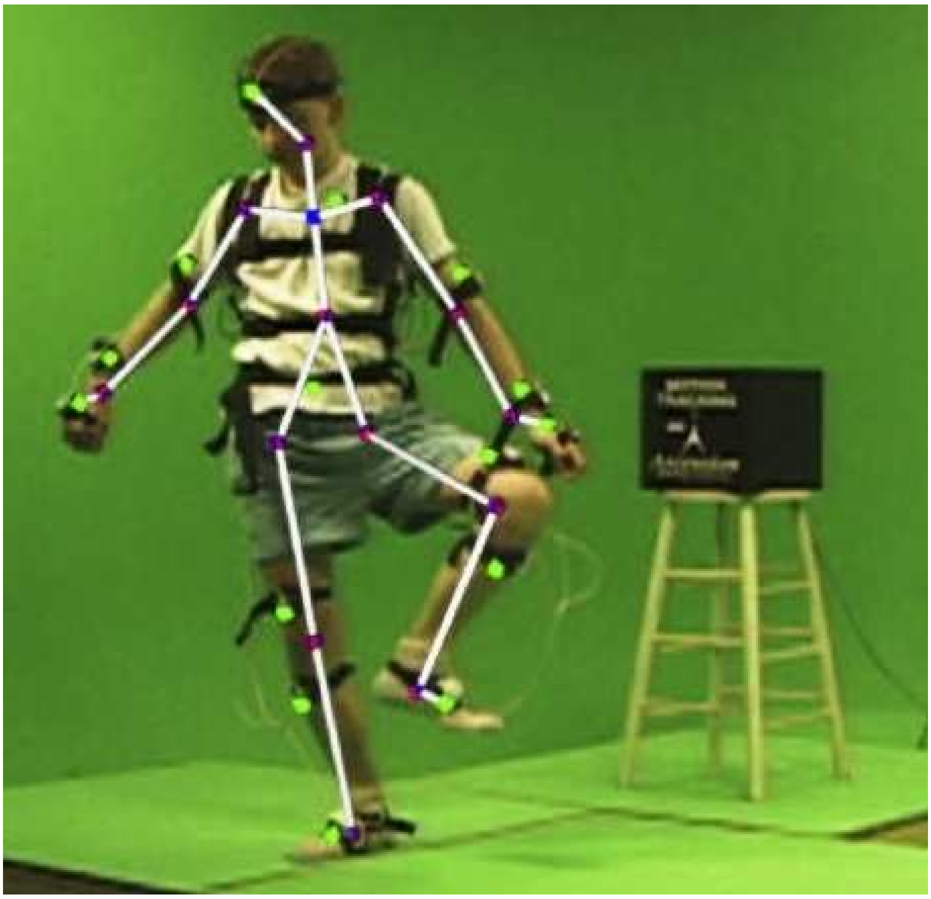


Figure 2.5: Test subject and generated model. The subject is wearing the motion capture equipment during a capture session; the superimposed skeletal model is generated automatically from the acquired motion capture data. The chest and pelvis sensors are located on the subjects back [4].

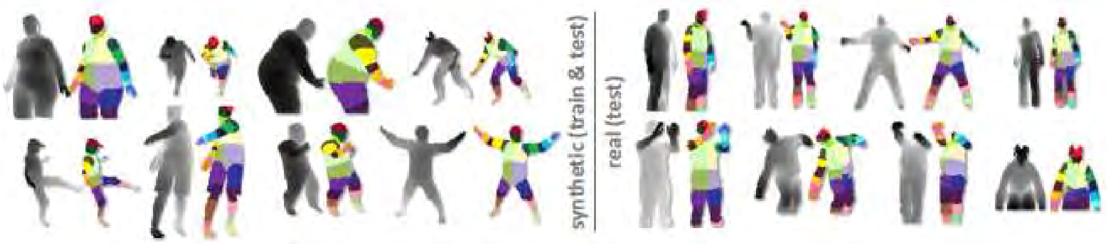


Figure 2.6: Synthetic and real data. Pairs of depth image and ground truth body parts [5].

Once formed, the memory matrix is used to classify/associate future inputs with those in the training set. The AMNN takes new or unseen inputs and correlates them with the memory matrix to determine the closest match to a known input type and declares the new input to be of that known type. This is useful for pose recognition because even though the general pose may be similar from one dismount to another, the exact positioning of joints will vary due to body types and personal preferences. AMNNs are very good at ignoring the effects of noisy inputs or variations from the training sets. AMNNs tolerance for noise and varience are the primary reasons it was pursued in this thesis. For classification the memory matrix and the class labels are the only pieces of information needed, making it easily transitioned to various platforms.

III. Methodology

This chapter details the methodology of the data collection and processing. This includes the process of converting raw data into a usable data base, achieving separability in the data, and defining the different poses (i.e. standing versus kneeling). Finally, a breakdown of the classification process is discussed.

3.1 Data Collection

3.1.1 MySkeleton Program. Utilizing the software development kit (SDK), made available by Microsoft, a program was created called MySkeleton. It uses Windows 7 drivers and the natural user interface (NUI) and the application programming interface (API), to support image and device management features. These features include access to the Kinect [3] sensors, image and depth stream data, and the delivery of processed image and depth data to support skeleton tracking.

Color and depth streams are a succession of frames collected and sent from the Kinect [3]. A frame is the information contained within a single still image of a particular type (e.g. depth, video, skeleton). The internal pipeline in the Kinect SDK runtime processes depth-frame data to generate skeleton frames. A skeleton frame event is determined by the API every time it processes a depth frame, even when no skeleton is detected in the scene.

Kinect [3] produces a video stream that is a depth image of its field of view and designates up to two dismounts by using different colors. Depth is shown in the form of a color scale, the lighter the color the closer the area is to the sensor. An image displaying the estimated joint locations and skeleton frame of the dismounts being tracked is shown. Every joint is mapped, therefore to achieve the highest level of accuracy, dismounts in their entirety need to be visible to the sensor. The Kinect also produces a color VGA display of the scene. A sample of how the information is displayed in the MySkeleton program is shown in Fig. 3.1. The data output from the MySkeleton program is a text file, containing labeled joint position estimates as three dimensional decimal values, which can be utilized by other programs [26].

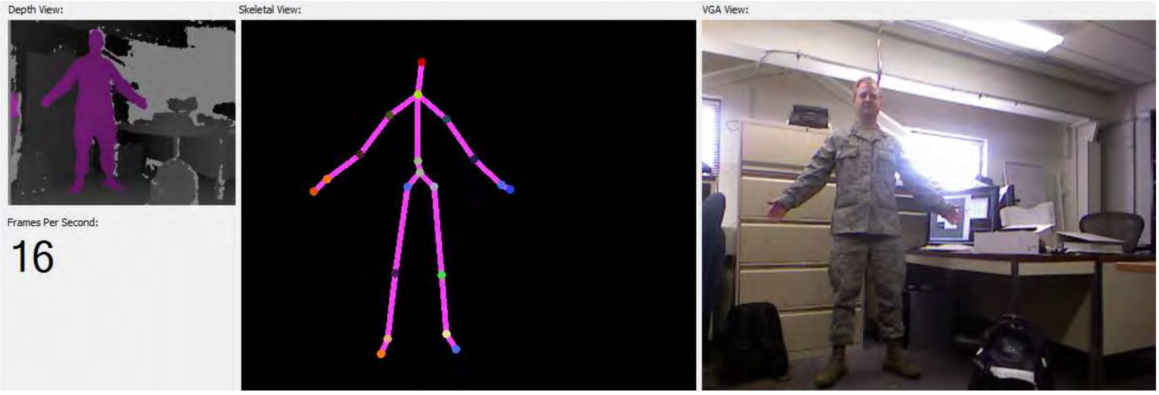


Figure 3.1: Sample view of data collection process using MySkeleton. Left frame is a depth view of the scene with shades of grey indicating depth of background and colored portion indicates the dismount. The center frame displays the joint location estimates as a connected skeleton. The right frame is a color VGA display of the scene.

3.1.2 Pose Collection. In order to collect data on the desired poses five steps are followed:

- the dismount is placed in the field of view of the sensor,
- the desired pose is described to and assumed by the dismount,
- the MySkeleton program is run to acquire joint estimates from Kinect sensor,
- the MySkeleton program is closed to generate the output file,
- the output file is renamed and saved for later use.

The above process is repeated until an output file has been saved for each pose. A list is maintained separately to match each collection with a brief textual description of the pose contained in each sample. The poses are all oriented so that they will be in a straight on view to the sensor. This operates on the assumption that the sensor is located at or near the intended target of the dismount and enables the collection of a wider variety of poses instead of the same poses with multiple orientations. Also, the side view may be simulated by swapping the x and z values of the data. The results of the pose collection process are detailed in Section 4.2.

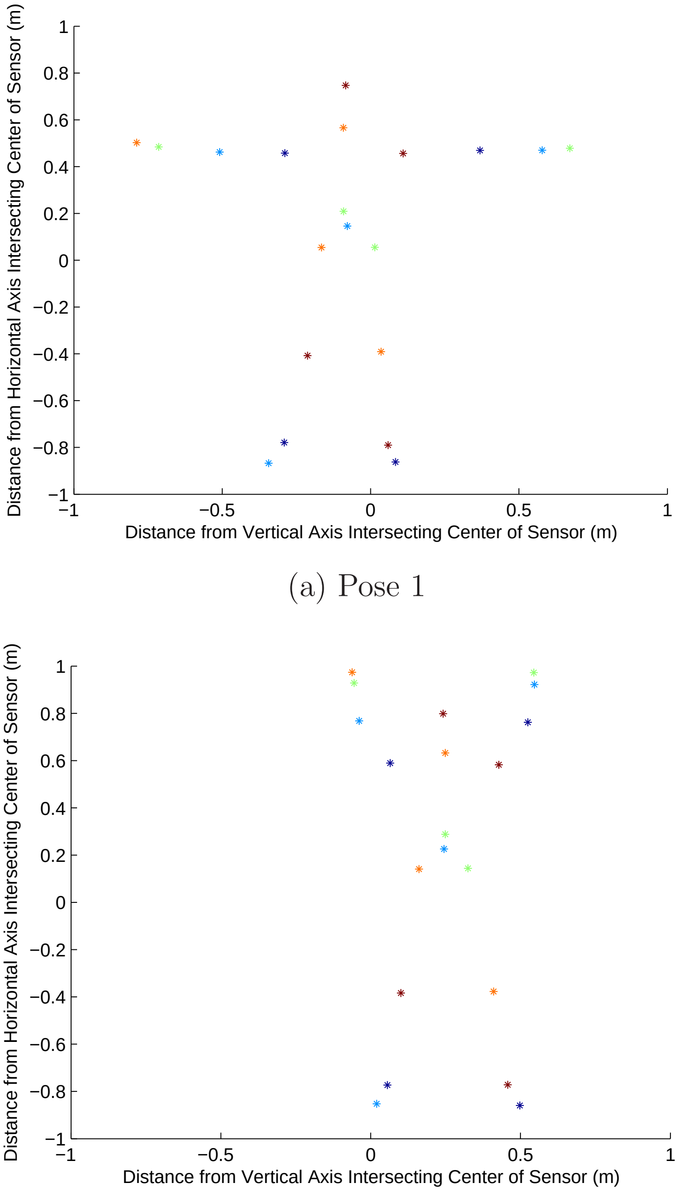
3.1.3 Pose Considerations. The data collection process utilized assumptions of AMNN functionality. It is unnecessary to collect and train for poses at every possible angle and position since an AMNN automatically assigns new inputs to their closest match contained in the training data. An example of this is demonstrated in Fig. 3.2, where the arms in Fig. 3.2 (a) are approximately orthogonal to the arms in Fig 3.2 (b). When a test sample is observed, and all joint positions are the same as Fig. 3.2 (a) and (b) except the arms, the network will map that pose to the closest match between the pose in Fig. 3.2(a) or (b) based on the actual positions of the arms.

Another consideration for collecting training poses is the aspect of being right handed or left handed. Due to the functionality of AMNNs, if a pose is recorded and trained into the network from a right handed subjects posturing, then an input of the same type of pose but in a left handed manner will not be recognized as the same threatening pose. Therefore, poses of the same type are collected based on dominant hand or dominant leg, and trained into the network separately as type right or left. An example of a right handed pose versus a left handed pose, when drawing a pistol from a waist holster, is shown in Fig. 3.3.

Along the same lines as the right and left handed issue is the standing pose versus the kneeling pose. Many standing poses have a pairing to a kneeling version of the same pose. From an automated recognition viewpoint, this places a large discrepancy in the data values as the vertical component values become closer together. To reduce the confusion of standing or kneeling on the AMNN, the standing and kneeling samples are grouped separately and only combined as needed during the training and evaluation stages. The overall impact of this separation on accuracy is discussed in Section 4.3.3.

3.2 Data Preprocessing

The data output by MySkeleton is given in decimal form up to six significant digits with three pieces of information for each data point and a label. The first two

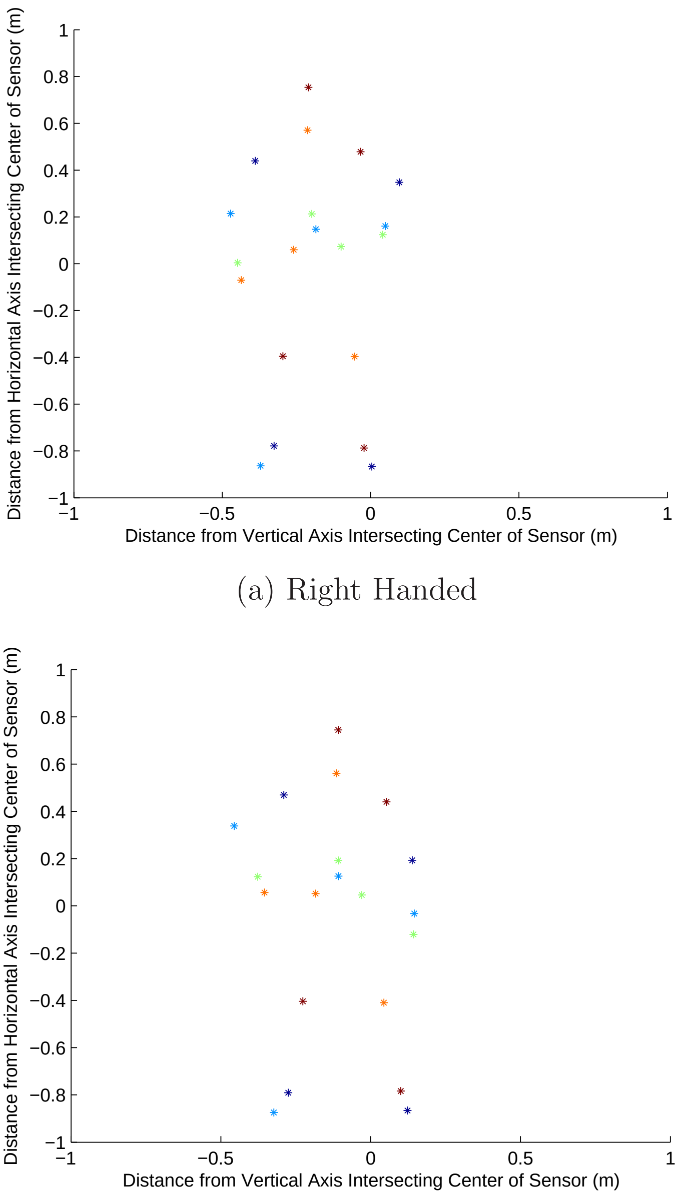


(a) Pose 1

(b) Pose 2

Figure 3.2: Joint position estimates extracted from Kinect. Example of collecting orthogonal poses to train the AMNN. The positions of the arms in (a) are approximately orthogonal to those in (b). Keeping all other joint positions the same, any new data with arms located somewhere between these two will be mapped to the closest set.

pieces of information, referred to as the x and y coordinates, are the horizontal and vertical distances of the joints relative to the center of the field of view. The third



(a) Right Handed

(b) Left Handed

Figure 3.3: Joint position estimates extracted from Kinect. (a) A right handed pose where the dismount is drawing a pistol from a holster at their waste. (b) A left handed pose where the dismount is drawing a pistol from a holster at their waste. Both the right handed and left handed versions of poses are collected and used to train the network.

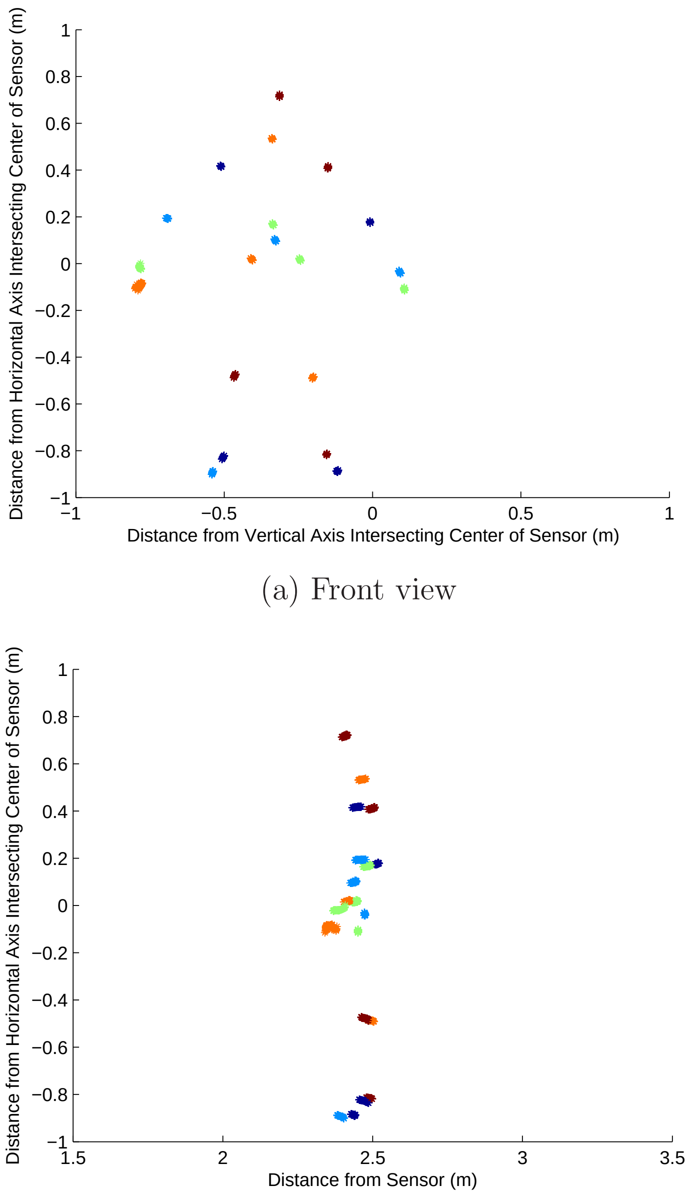
piece of information, referred to as the z coordinate, is the depth/distance from the sensor. Each set of x, y, and z coordinates are given a label to indicate which joint they represent. The labels are numerical from 0 to 19 and correspond to joints/body parts as listed in Table 3.1.

Table 3.1: Labels and associated joints when output from the MySkeleton program.

Numbered Label	Joint Represented
0	Groin
1	Abdomen
2	Neck
3	Head
4	Right Shoulder
5	Right Elbow
6	Right Wrist
7	Right Hand
8	Left Shoulder
9	Left Elbow
10	Left Wrist
11	Left Hand
12	Right Hip
13	Right Knee
14	Right Ankle
15	Right Foot
16	Left Hip
17	Left Knee
18	Left Ankle
19	Left Foot

3.2.1 Data Simplification. Each output file has multiple x, y, and z values for each individual joint position due to the temporal range and sampling rate of each sample. The result of these multiple values for each joint is shown in Fig. 3.4. This thesis is focused on recognizing static poses, therefore, the data for each joint is averaged to yield a single well defined estimate of each joint position as shown in Fig. 3.5. In both Fig. 3.4 and Fig. 3.5, the (a) plot is the front view illustrating the x and y coordinate data, while the (b) plot represents the z and y data.

3.2.2 Normalization. As previously stated the data values are the relative distances of the joints based on the distance and field of view of the sensor. This relative distance value can introduce variance in the data if the subject of interest

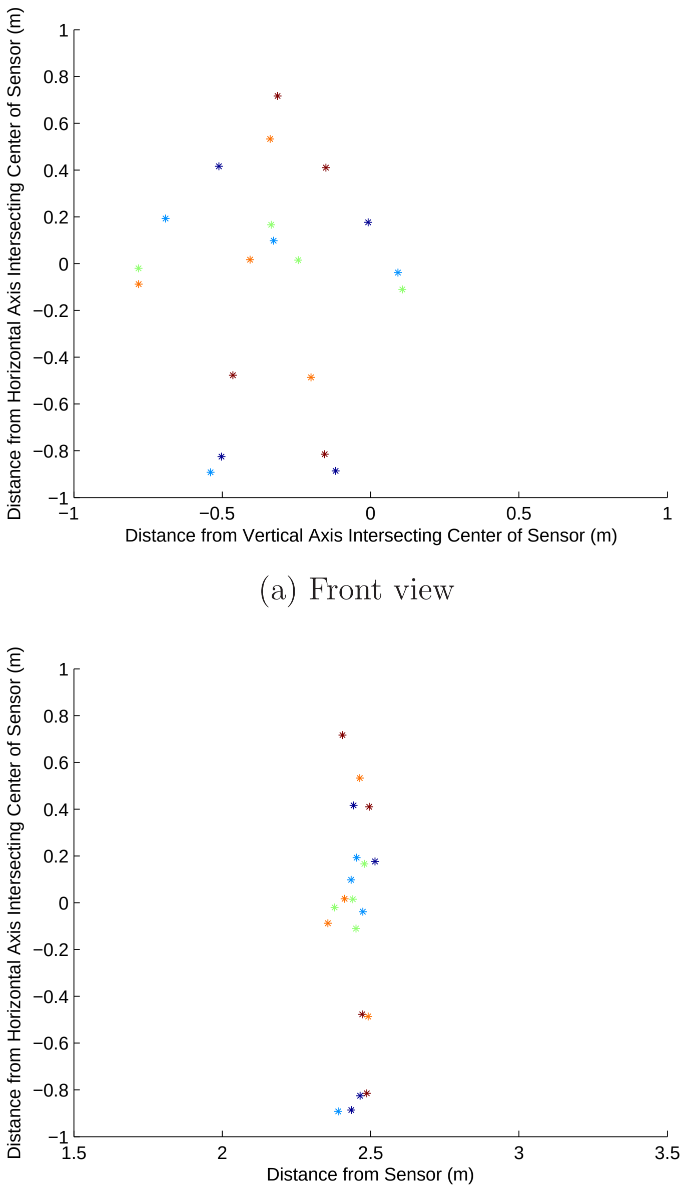


(a) Front view

(b) Side view

Figure 3.4: (a) Frontal view of data extracted from the sensor (x,y). (b) Side View of data extracted from the sensor (z,y). Both views are using unprocessed data in its original form

is not standing in the same location as the trained samples, thus normalization is necessary. Normalization is accomplished by subtracting the respective x, y, and z, values of the groin from all the joints, causing the groin location values to become (0,0,0) and the rest of the joint positions to be relative to the groin. This normalization



(a) Front view

(b) Side view

Figure 3.5: (a) Frontal view of averaged data (x,y). (b) Side View of averaged data (z,y). Both views are displaying the end result of noise and variance reduction through averaging.

allows the data to be analyzed and compared regardless of where the test subject was located within the sensor field of view.

3.2.3 Creating Separability. Previous work with associative memory neural networks using binary input values yields results with a high tolerance for noise [27]. Therefore, the data is converted to binary equivalent values. To remove the need to manage sign bits, the data is shifted equally to become positive. This shifting is achieved by adding a predetermined value, in this case 1.5m, to all the data values. Since the normalizing point is the subjects groin, and human appendages are typically equal to half their overall height, a value of 1.5m ($\approx$5ft) accounts for human attributes and includes a buffer.

Since binary values only represent whole numbers, the data is scaled by a factor of 10,000. This factor preserves the data while enabling its conversion to binary. In order to achieve greater separability, and yield a high level of accuracy, the data is converted from binary values of 1's and 0's to bipolar values of 1's and -1's. A comparison of the results using binary versus bipolar values is shown in Section 4.3.2. By treating each character of the bipolar representation as a dimension, the conversion to bipolar values effectively projects the three dimensional data into 45 dimensional space, a 15 bit binary value for each of the original x, y, and z coordinates.

3.3 Training the Associative Memory Neural Network

Once the data is processed into a usable form, the AMNN is trained. As mentioned in Section 2.2.3, AMNNs require input vectors and the associated output vectors, $\dot{a}$ and $\dot{b}$ respectively. The input vectors $\dot{a}$ are formed through use of concatenation on the bipolar data stored as 20x45 matrices per pose initially and becoming 900x1 vectors. The output vectors $\dot{b}$ are binary, consisting of 0's and a single 1 corresponding to the pose its respective input vector represents. If the training set contained only one sample of each pose then stacking the output vectors next to each other would form an identity matrix. A sample output vector which would correspond to the second input is shown in Eq. 3.1.

$$\dot{b}_2 = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \end{bmatrix}^T \tag{3.1}$$

The next stage required creation of a weight matrix for each input pose represented by $\ddot{W}(k)$. The k represents an individual training pattern or the k^{th} pose. The notation $\dot{a}_k$ and $\dot{b}_k$ are used to represent the input and output column vectors associated with that pattern. Eq. 3.2 gives the general form for creating the individual weight matrices, while Eq. 3.3 and Eq. 3.4 expand it into a more explicit view.

$$\ddot{W}(k) = \dot{b}_k * \dot{a}_k^T \tag{3.2}$$

$$\ddot{W}(k) = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix} \tag{3.3}$$

$$\ddot{W}(k) = \begin{bmatrix} b_1a_1 & b_1a_2 & \cdots & b_1a_n \\ b_2a_1 & b_2a_2 & \cdots & b_2a_n \\ \vdots & \vdots & \vdots & \vdots \\ b_ma_1 & b_ma_2 & \cdots & b_ma_n \end{bmatrix} \tag{3.4}$$

Where $\ddot{W}$(k) is the weight matrix associated with the k^{th} input and output, $\dot{a}_k$ and $\dot{b}_k$ are the k^{th} input and output vectors, a_1 through a_n are the individual elements of the input vector $\dot{a}$, and b_1 through b_m are the individual elements of the output vector $\dot{b}$. n and m are the lengths of the input and output vectors respectively.

The completion of the weight matrices enables the final stage of the training process in which the memory matrix is calculated. The memory matrix is a summation of all the weight matrices, as indicated in Eq. 3.5 where p is the number of distinct input patterns. The memory matrix can also be computed piecewise or expanded using Eq. 3.6. The matrix $\ddot{M}(0)$ is defined as the zero matrix containing no memory. The main difference between Eq. 3.5 and Eq. 3.6 is that Eq. 3.5 is a batch learning process and creates the memory matrix all at once, while Eq. 3.6 is updated with each new input. When k is equal to the number of training inputs both forms of the equation yield the same result.

$$\ddot{M} = \sum_{k=1}^{p} \ddot{W}(k) \tag{3.5}$$

$$\ddot{M}(k) = \ddot{M}(k-1) + \ddot{W}(k) \tag{3.6}$$

Where $\ddot{M}$ is the full memory matrix, $\ddot{M}$(k) is the memory matrix up to the k^{th} input, p is the number of inputs, and $\ddot{W}$(k) is the weight matrix associated with the k^{th} input and output.

The final step in using an AMNN is utilizing the memory matrix to classify any new or previously observed input patterns. This is accomplished through a process referred to as recall, where the input vectors are assumed unit length, shown in Eq. 3.7 [25]. The memory matrix is multiplied with an appropriately formed input vector to create an output vector. The location/index of the highest value within the output vector indicates which trained pose the input most closely resembles. For example, if the vector in Eq. 3.8 is the result of an input vector times the memory matrix then it would identify the input as being of type 2 since the highest value, 9 in this case, is located in the second position of the vector.

$$\hat{b} = \ddot{M}\dot{a}_i \tag{3.7}$$

$$\hat{b}_2 = \begin{bmatrix} 2 & \mathbf{9} & 3 & 4 & 1 & 5 \end{bmatrix}^T \tag{3.8}$$

Where $\ddot{M}$ is the memory matrix, $\dot{a}_i$ is an input vector, $\hat{b}$ is the estimate of a trained output, and $\hat{b}_2$ is an example of an output that is classified as the second training type.

3.4 The Network

As the memory matrices are created it is necessary to build a framework for their use. Any newly observed data, for the purposes of this thesis, is assumed to be collected using the Kinect [3] sensor and the MySkeleton program. The first stage is to read in the data, average it, and normalize it. Then the data is shifted to become positive and multiplied by 10000 to preserve the decimal values. Each x, y, z value is then converted to its 15 bit binary representation and converted to bipolar values. The data is then concatenated to transition from a matrix of values to a single vector.

Once vectorized, the data is applied to the memory matrix to form a recall vector. The index of the maximum value in the recall vector determines the pose most likely observed. Finally, the index is checked against the pose labels and classified as threatening or non-threatening. The entire post collection threat determination takes less than a second on a computer with a 3GHz processor. The accuracy of this process and the validation techniques are discussed in Chapter IV.

IV. Experimental Results and Analyses

This chapter documents the testing of proposed theories and methods throughout this thesis and the experimental results they produced. A brief description of how accuracy was measured and reported and a description of the data sets used during the training and testing processes is given. The effects of various parameter and method choices on overall accuracy is detailed. Finally, an accuracy estimate for the end classifier and its tolerances is shown.

4.1 Determining Accuracy

This thesis is focused on determining whether a dismount is threatening or not based on their posturing. Therefore any accuracy rating should indicate how often a correct determination was made, and any acceptable tolerances. One method of testing whether the system is working or not involves training and testing on the same data set. When done properly, with well formatted data, the network will recognize and correctly identify each pose.

Another method of measuring accuracy is to view the networks tolerance to noise. Since the network is utilizing binary/bipolar inputs a noise effect is considered anything that would cause a bit to be considered opposite of its true value when the information reaches the processing stage. Noise effects are applied as a percent of total bits, from 1% to 100% in single percentage increments, changed from their original values. The location of the flipped bits is randomly selected at each noise level and the process is repeated 100 times to discern a mean noise tolerance. See also Fig. 4.2 for a sample of how the results are displayed upon completion.

Since the purpose of this thesis is to provide a threatening determination from a pose an alternate measure of success is available. It is possible to consider the network successful even in the event that a wrong pose was identified but it was still of the correct threat determination. In several cases the poses most often confused with each other are of the same threat level and yield a correct threat assessment regardless of the actual pose being incorrect. As illustrated in Fig. 4.1 the overall accuracy for threat assessment remains high longer than the pose accuracy and never drops completely to zero even in the presence of extremely high noise.

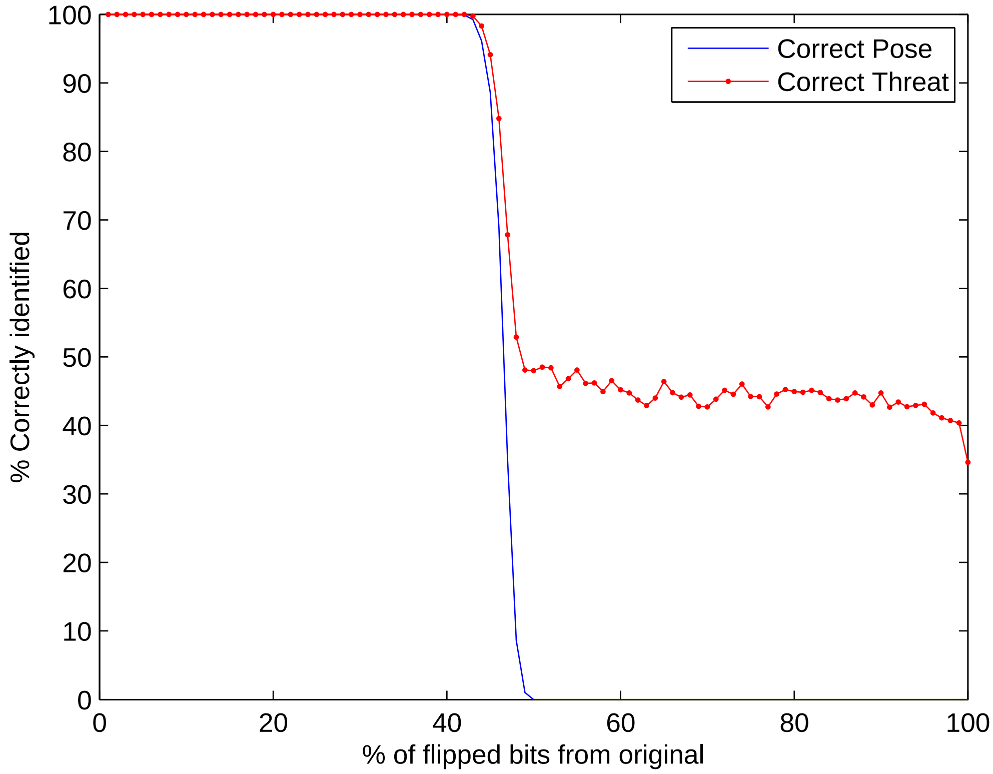


Figure 4.1: Example of accuracy when considering correct pose identification versus correct threat assessment. This is based on training and testing the network with the same set of data.

4.2 Data Used

The data used for training and testing purposes throughout this thesis was obtained through the collection process described in Section 3.1.2. A total of five sets of data were collected across two dismounts. A set of data is considered to be a collection of joint information for each pose of interest amounting to 26 unique poses per set. A list of the poses contained within each set, their associated number, and threat determination, is shown in Table 4.1 with a graphical view of each pose shown in Appendix A. The only threat labels that could easily be interpreted differently are for poses 10 and 11, where the dismount is standing with a rifle in a carry position in front of them. These two poses were labeled as non-threatening because it is the assumed manner in which friendly military dismounts are viewed in scene and reduces false positives. Another assumption is that a hostile dismount carrying a rifle seeks to conceal their weapon as long as possible and never achieve the posturing described by poses 10 and 11.

Table 4.1: Text description of poses collected, their numeric references, and associated threat determination for each.

Label	Threat	Description
1	No	Standing, arms down and held slightly away from sides
2	No	Standing, arms straight up past the head
3	No	Standing, arms straight out to the side
4	Yes	Standing, drawing a pistol, waist holster, dominant right hand
5	Yes	Standing, drawing a pistol, waist holster, dominant left hand
6	Yes	Standing, aiming a pistol, dominant right hand
7	Yes	Standing, aiming a pistol, dominant left hand
8	Yes	Standing, aiming a rifle, dominant right hand
9	Yes	Standing, aiming a rifle, dominant left hand
10	No	Standing, carrying a rifle in front of body, dominant right hand
11	No	Standing, carrying a rifle in front of body, dominant left hand
12	No	Standing, waving with right hand in air, left arm at side
13	No	Standing, waving with left hand in air, right arm at side
14	No	Standing, reaching for handshake with right arm
15	No	Standing, reaching for handshake with left arm
16	No	Standing, arms crossed in front of chest
17	No	Standing, hands behind head in surrender
18	No	Standing, arms straight out in front of themselves
19	Yes	Standing, throwing grenade with right hand
20	Yes	Standing, throwing grenade with left hand
21	Yes	Kneeling, aiming a pistol, dominant right hand
22	Yes	Kneeling, aiming a pistol, dominant left hand
23	Yes	Kneeling, aiming a rifle, dominant right hand
24	Yes	Kneeling, aiming a rifle, dominant left hand
25	No	Kneeling, hands behind head in surrender
26	No	Kneeling, arms open as if to hug a child

Table 4.2 lists some characteristics about the dismounts involved in the data collects and the data sets they are associated with. The subjects height is relevant

Table 4.2: Test subject attributes and associated data sets

Subject designation	Height (ft)	Data Sets
Male1	5'9"	Set1, Set2, Set3
Male2	6'2"	Set4, Set5

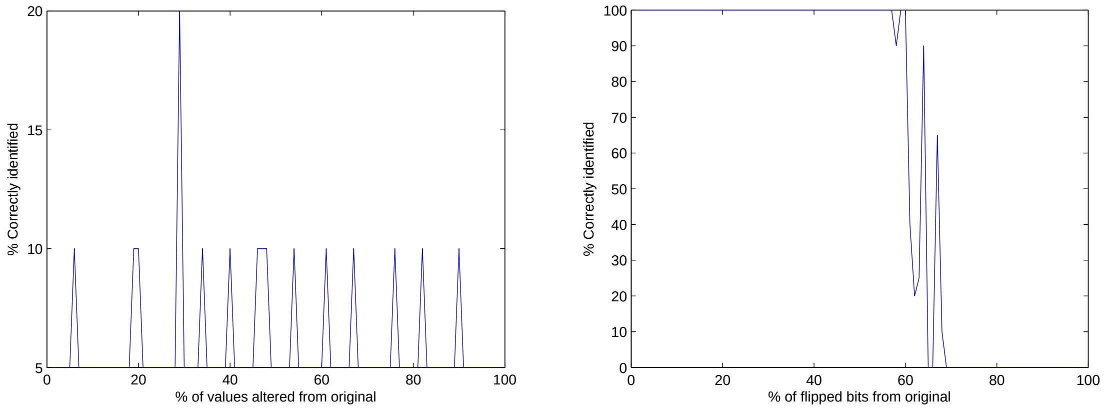
due to its effect on joint relationships. The spacing of a dismounts joints will be different based on their height and body shape.

4.3 Effects of Parameter and Method Choices

Various parameter and methodology choices are employed, the results of these choices are geared toward improving the accuracy and applicability of the system as a whole. The following sections explain and expand upon what these choices are and their impact.

4.3.1 Decimel versus Bipolar. As mentioned in Section 3.2.3 a determination was made to convert the inputs from their original decimal form to a bipolar representation. This had a major impact on the initial results as it provided much needed separability. At the preliminary stages of training the decimal form data showed anywhere from 5% - 20% accuracy when testing the network with the same data it was trained with. However, when the data is converted to the bipolar form, the network produces a 100% accuracy rating. As shown in Fig. 4.2 the conversion to bipolar values causes a substantial increase in accuracy and noise tolerance.

4.3.2 Bipolar and Binary Values. The conversion of the data to binary/bipolar values has a clear and significant impact on the overall accuracy of the network. In an effort to produce the best results the effects of using binary/bipolar values for the inputs as well as the predefined output matrices are explored. This results in four different test cases: binary inputs with binary output, binary inputs with bipolar output, bipolar inputs with binary output, and bipolar inputs with bipolar output. To allow direct comparison each test case is run using the same set of data to train and test the respective networks. Fig. 4.3 illustrates the effects of these



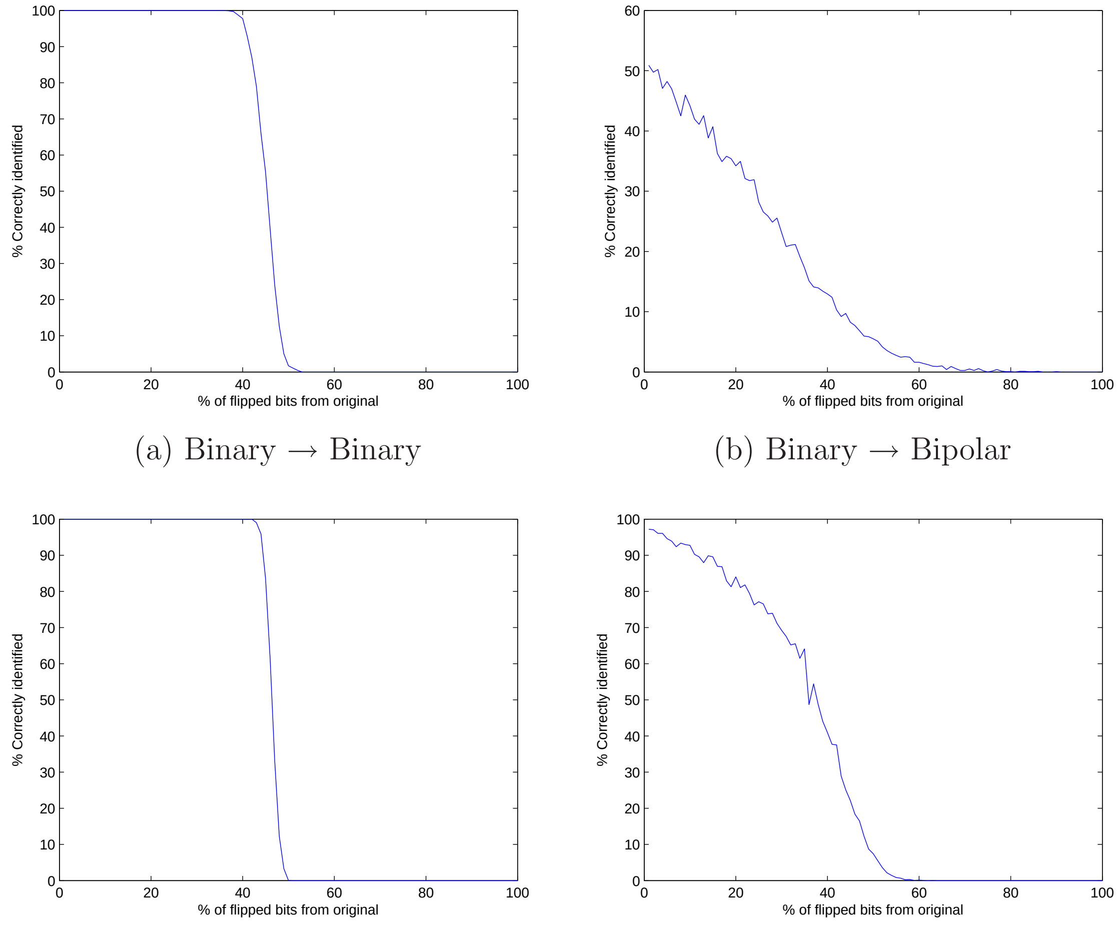
(a) Decimal Form

(b) Bipolar Form

Figure 4.2: Pose identification accuracy as the amount of data affected by noise is increased. (a) shows the accuracy when using unprocessed decimal data. (b) shows the accuracy when using data that is converted to a bipolar representation. In both (a) and (b) the data used to train the network is also used to test the network.

four cases. The two cases using binary values in the output matrix, Fig. 4.3(a) and Fig. 4.3(c), clearly outperform the bipolar output cases, Fig. 4.3(b) and Fig. 4.3(d). The combination yielding the best results is the one with bipolar input values and a binary output matrix, case (c) in Fig. 4.3. The bipolar inputs enable a wider range of values in the weight matrices while the 0's of the binary output matrix help to reduce crosstalk, the effect one input has on another. Note that the results displayed in Fig. 4.3 are created using the same data set to train and test the network and yield higher accuracies than what is normally seen.

4.3.3 Standing and Kneeling Separation. Separation of standing and kneeling poses proved to be unnecessary. This result came about after making the final determination of using binary values to create the output matrix. The 0's in the output matrix aid to remove and almost eliminate the presence of crosstalk within the network. As a result, the large degree of variance between the standing and kneeling poses does not skew the memory matrix. In contrast, if a bipolar output matrix is used the difference between the standing and kneeling poses heavily disrupts the memory matrix. A comparison of the two output matrices affects is shown in Fig. 4.4. As seen



(a) Binary → Binary

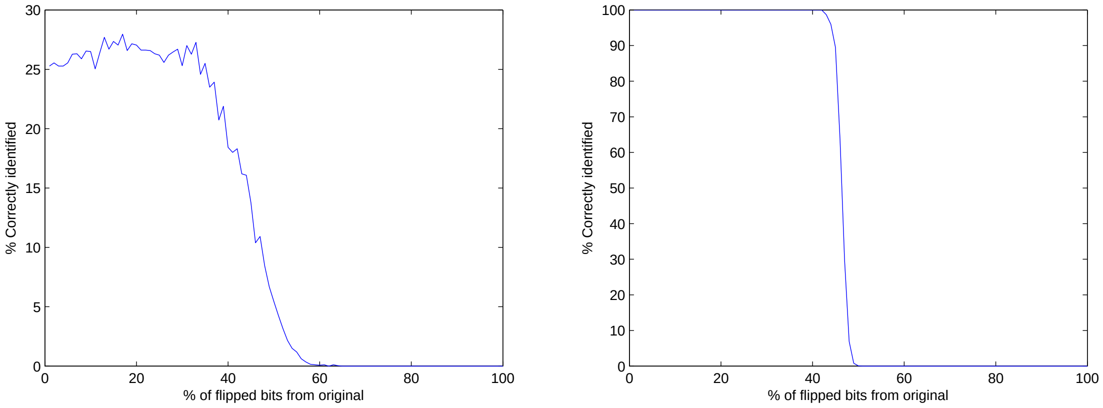
(b) Binary → Bipolar

(c) Bipolar → Binary

(d) Bipolar → Bipolar

Figure 4.3: Comparison of results when using binary/bipolar inputs and outputs. The notation labeling the graphs is as follows: x → y represents data of type x is the input paired with an output of type y. Case (c) uses bipolar inputs and a binary output matrix to train the network, and yields the most favorable results.

in Fig. 4.4(a) the initial accuracy is very low in the bipolar case but the overall drop off rate remains similar to those seen when standing and kneeling poses are trained and tested separately. The results in Fig. 4.4(b) for the binary case maintain a high level of accuracy with a sharp decline around 50%. In both cases the same set of data is used to train and test the network.



(a) Bipolar Output

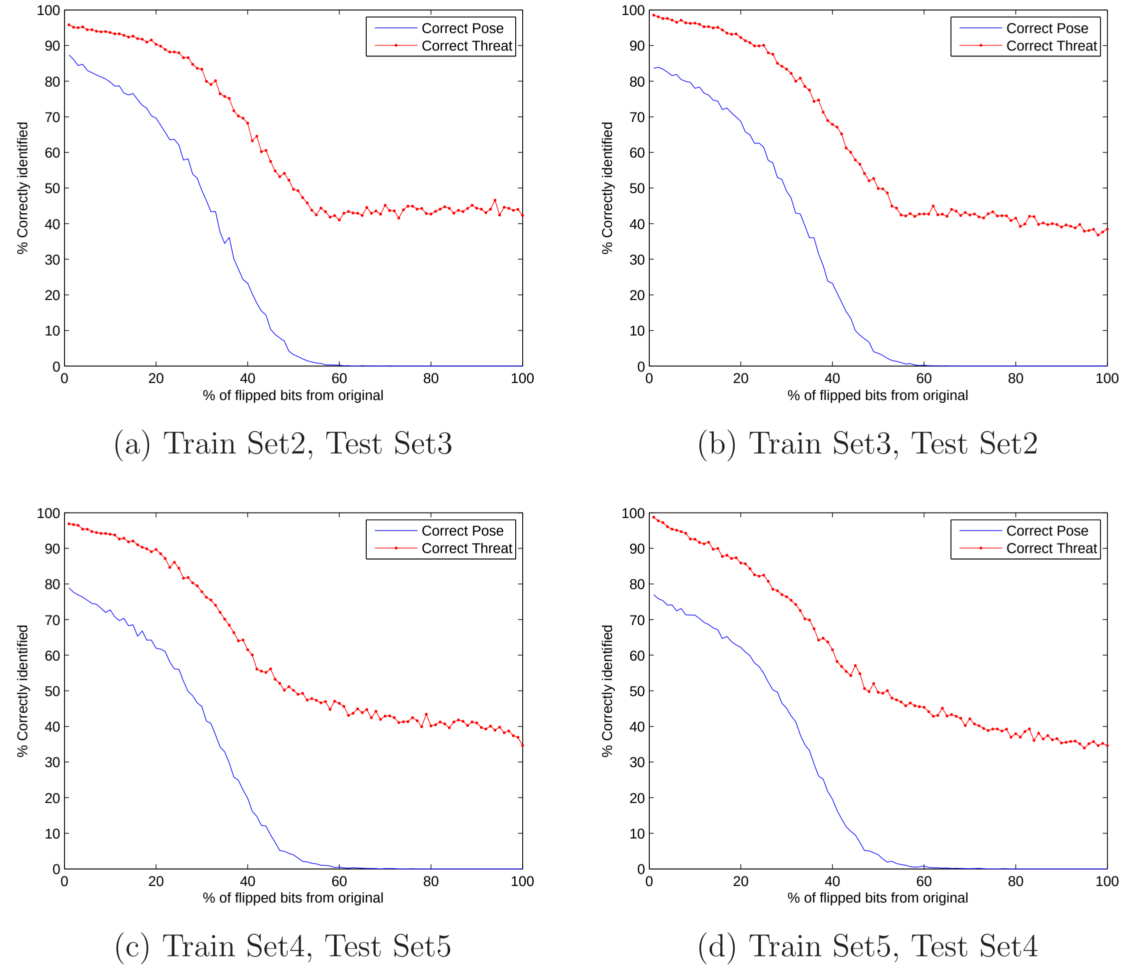
(b) Binary Output

Figure 4.4: Accuracy levels obtained when the network is trained and tested with a set of data containing both kneeling and standing poses. In (a) the output matrix is defined with bipolar values. In (b) the output matrix is defined with binary values. The results in (b) indicate that it is unnecessary to separate the standing and kneeling data when a binary output matrix is used.

4.4 Classifier Accuracy

When collecting data, Set1 is taken independant of all other data, while the others are collected in the following pairs: Set2 and Set3, and Set4 and Set5. This yields data sets that are very similar to each other and produces high levels of accuracy when trained and tested together, as can be seen in Fig. 4.5. The results of training the network with a single set of data and then testing it with a set not closely related are less favorable and can be seen in Fig. 4.6. In Fig. 4.6(a) the two sets are collected at different times but with the same subject. This situation yields more favorable results than Fig. 4.6(b) where the sets used to train and test the network are created from different dismounts.

In order for the system to be useful it must be able to map seen and unseen inputs to the appropriate threat or pose, thus the method used for validation is that of K-fold cross validation. In K-fold cross-validation, the training data is divided into K approximately equal sized groups, in this case 5 data sets. One of the sets is withheld from the training step and is used as the test set. The other K - 1 sets are used to



(a) Train Set2, Test Set3

(b) Train Set3, Test Set2

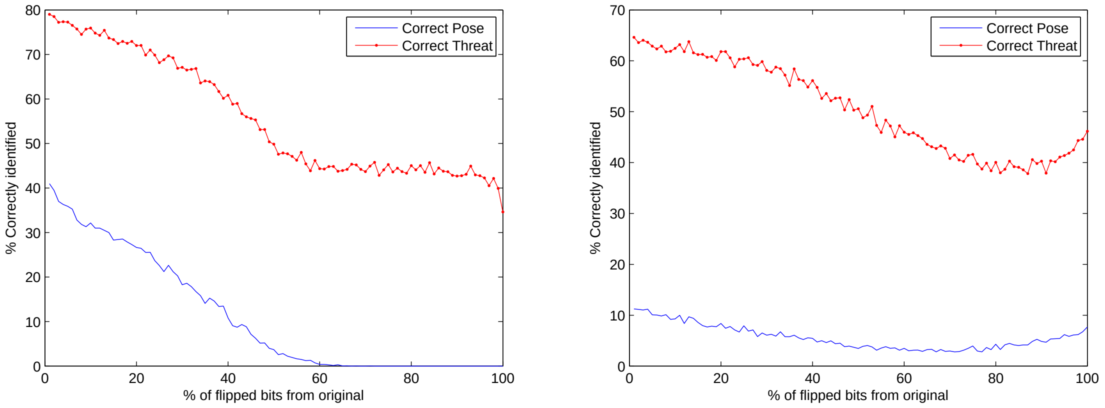
(c) Train Set4, Test Set5

(d) Train Set5, Test Set4

Figure 4.5: Accuracy levels obtained when network is trained and tested with sets of data that are very similar to each other. The dismount observed in Set2 and Set3 is different than the dismount in Set4 and Set5.

train the network, and this process is repeated for each of the possible K - 1 sets. The result is an estimate for the systems accuracy when applied in a real world situation.

Since the order of training has no effect on the outcome, a K = 5 yields five unique combinations of data sets for the K-folds validation. The results of these five sets and their averaged output are shown in Fig. 4.7. Fig. 4.7(e) illustrates the network viewing an unknown condition. The network of Fig. 4.7(e) is trained with Male1 through Set2 and Set3, but Set1 is a unique data set. The collection of Set1



(a) Train Set1, Test Set3

(b) Train Set2, Test Set4

Figure 4.6: (a) Accuracy levels obtained when network is trained and tested with disimilar sets from the same dismount. (b) Accuracy levels obtained when the network is trained with a set from one dismount and tested with a set created by a different dismount.

occurred weeks before the other two and no effort is made to mimick Set1, since each data set is intended to be unique and provide as realistic data as possible.

It is useful to know what poses are being most often confused with each other. This allows measure to be taken to achieve greater separation, as well as letting the user know if the network has a chance to declare false positives with threat determinations. Table 4.3 is the confusion matrix for indicating which poses are identified correctly and incorrectly. Table 4.4 is the confusion matrix illustrating the likelihood that a threat determination, regardless of pose, is correct. The rows are the known input classes and the columns indicate what the network classifies the inputs as. The abbreviations used and their description are:

- PA is the producers accuracy, the percentage of samples that belong to a given class that are actually classified as that class,
- OE is the omission errors, percentage of samples belonging to the given class but omitted,
- CA is the consumers accuracy, percentage of what is labeled as belonging to a given class and is actually a member of that class,

- CE is the commission errors, percentage of samples classified as a given class not belonging to that class,
- Kappa is the kappa statistic, a value bounded between -1 and 1 indicating the level of agreement between actual classes and predicted classes.

In Table 4.3 the PA and CA values fluctuate between 0% and 100%. This indicates that some poses are unique enough to be readily identified and others are often confused with other poses. In contrast, the PA and CA values in Table 4.4 are consistently high indicating high levels of correct threat discernment. A closer inspection of Table 4.3 reveals that the poses which get commonly confused with each other are the ones involving weapons. Since threat is being viewed in a binary manner of threatening or non-threatening, this confusion is acceptable. Another point worth mentioning is that the kneeling rifle poses were rarely identified correctly and only chosen as the estimate once during the K-fold cross validation, indicating a lack of consistency for those poses in the training data or that their inclusion is not necessary given the standing versions of those poses. The Kappa statistics for both Table 4.3 and Table 4.4 are similar, 0.51 and 0.61 respectively. This similarity is expected since both tables are based on the same data. The kappa values suggest that the model is in moderate to good agreement with the truth data.

4.5 Analysis of Results and Summary

The accuracy of the system is determined in two ways: through correct identification of the pose presented, and making a correct threat determination. The effects of using binary versus bipolar values for the input values and the output values are explored. The results are that the best levels of accuracy are obtained when the inputs are bipolar and the output matrix is binary.

The separation of the standing and kneeling poses is unnecesary as the crosstalk is reduced through the use of the binary output matrix. If the bipolar output matrix is desired then separation does have a significant impact on system accuracy. The

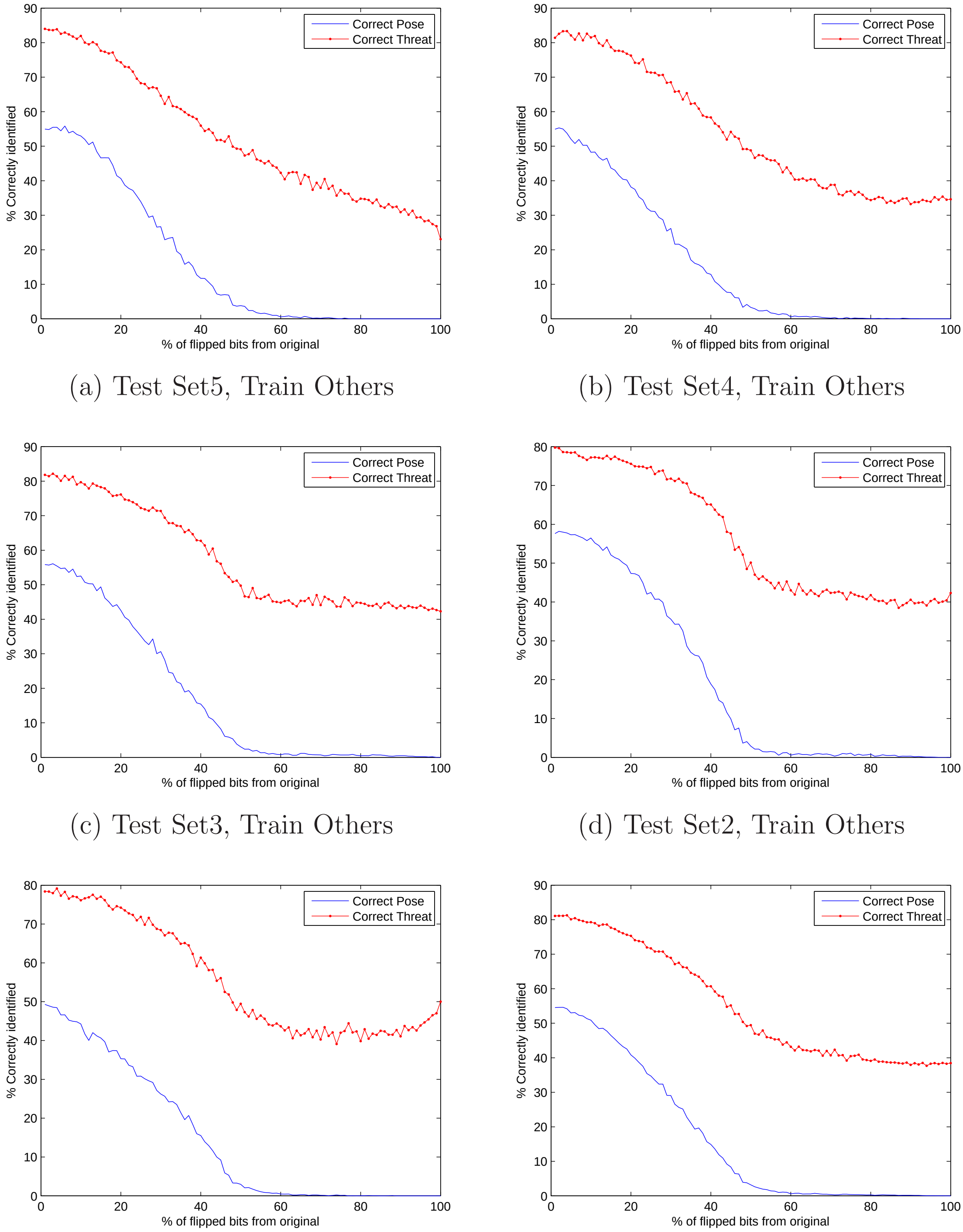
Table 4.3: Confusion matrix based on correct pose identification. The producers accuracy (PA) and consumers accuracy (CA) vary with each pose, indicating that some poses are more distinct and easier to recognize than others. A Kappa statistic value of 0.51 indicates a high level of agreement between the actual and predicted classifications.

	Estimated Class																												
Truth	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	#Samples	PA%	OE%
1	4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	80	20
2	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	100	0
3	0	1	1	0	0	0	2	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	5	20	80
4	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	100	0
5	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	100	0
6	0	0	0	0	0	2	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	5	40	60
7	0	0	0	0	0	0	2	0	2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	5	40	60
8	0	0	0	0	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	5	0	100
9	0	0	0	0	0	0	1	0	3	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	5	60	40
10	0	0	0	0	0	0	0	0	0	4	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	5	80	20
11	0	0	0	0	2	0	0	0	0	0	0	0	2	0	0	1	0	0	0	0	0	0	0	0	0	0	5	0	100
12	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	100	0
13	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	5	100	0
14	0	1	0	0	0	0	1	0	0	0	0	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0	5	40	60
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	1	0	0	0	0	0	0	0	0	5	80	20
16	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	5	60	40
17	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	5	60	40
18	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	5	60	40
19	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	0	0	0	0	0	5	20	80
20	0	0	0	0	0	0	0	0	3	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	5	0	100
21	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	2	1	0	0	0	0	5	40	60
22	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	5	60	40
23	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0	0	0	1	0	0	0	5	20	80
24	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	0	1	0	1	0	0	0	0	0	5	0	100
25	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	5	20	80
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	5	100	0
Totals	6	8	2	6	9	7	11	1	8	5	1	6	9	3	7	8	6	6	2	2	5	5	1	0	1	5	69		
CA%	67	63	50	83	56	29	18	0	38	80	0	83	56	67	57	38	50	50	50	0	40	60	100	100	100	100		Kappa	
CE%	33	37	50	17	44	71	82	100	62	20	100	17	44	33	43	62	50	50	50	100	60	40	0	0	0	0		0.51	

Table 4.4: Confusion matrix based on threat determination. The producers accuracy (PA) and consumers accuracy (CA) are both high with a Kappa statistic of 0.61. This indicates a high level of agreement between the actual and predicted classifications.

	Estimated Class				
Truth	Non-Threat	Threat	Samples	PA%	OE%
Non-Threat	59	11	70	84.29	15.71
Threat	14	46	60	76.67	23.33
Totals	73	57			
CA%	80.82	80.70		Kappa	
CE%	19.18	19.30		0.61	

network is trained and tested with various combinations of data in an effort to discern the best way to achieve accurate results in an operational environment. For use in a limited environment, focused on watching one or two select dismounts, a small set of training data geared toward the dismount of interest will achieve very good results. In order to be used on a larger scale, a wide range of training data is preferred to expose the network to wider variety of body types and pose variations. The more versions of a pose that the network is trained with the more accurate the network will be.



(a) Test Set5, Train Others

(b) Test Set4, Train Others

(c) Test Set3, Train Others

(d) Test Set2, Train Others

(e) Test Set1, Train Others

(f) Average accuracy K = 5

Figure 4.7: K - folds cross-validation with K = 5. (a) - (e) are trained using all but the identified test set. (f) is the average of all cases.

V. Conclusions and Future Work

This chapter summarizes the methods used, and conclusions reached, as a result of this thesis effort. Potential avenues for future work are presented, and a few of the contributions this work yields to the field of dismount threat recognition are discussed.

5.1 Summary of Methods and Conclusions

The main focus of this thesis is to enable a computer system to observe and recognize dangerous situations. This is attempted through the use of associative memory neural networks (AMNN) and joint mapping methods developed by Shotton *et al* [5]. The process involved using a real time depth camera, the Kinect, to observe a dismount and provide joint position estimates through the method developed in [5]. The information is extracted from the Kinect using the software development kit provided by Microsoft and a separate program developed at AFIT. The joint position estimates are collected for several different poses performed by two individuals.

The data is then pre-processed to transition from the raw decimal numbers to a binary representation of the values. The binary data is converted one more time to bipolar to increase data separability, and used to train an AMNN. Several training and testing combinations are performed to discern the combinations that yield the best results, both initially and in the presence of noise. Accuracy is measured both in terms of correct pose identification as well as proper threat determination. The threat determination consistantly yields equal or better accuracies than the pose identification.

Overall, this thesis successfully proved the concept of the ability to automatically discern whether a situation is threatening or not. Using the results from Fig. 4.7(f), the best estimate for real world threat discernment accuracy is approximately 81%. This number is not entirely indicative of real-world accuracy as the testing data is limited in scope for this thesis. Methods to make the accuracy better is discussed in Section 5.2. Given the relatively high results for the threat discernment, automated pose recognition could be refined and deployed as part of a larger dismount threat recognition package or used alone as a means of early warning.

5.2 Future Work

This research effort proposed and tested a new method of discerning the threat level of a dismount. Although it is successful in proving the concept, there is more work to accomplish to improve, refine, or streamline the process.

A major area for improvement is to collect more training data. This data can be a greater number of unique poses for the network to learn and be ready for, as well as collection from a larger variety of body shapes. The general poses may appear the same to an observer, but the relative joint positions will vary to what is comfortable and familiar to the subject. Having a larger data set to train and test with would allow the accuracies generated to be more representative of a fielded system.

Improvements to the flow of the system can be made so that it will operate in real-time. The current process is not real-time because of a disconnect between the Kinect output and the data processing element. Currently the MySkeleton program runs the Kinect for a few seconds to generate joint position estimates, the program is closed to generate the output file, and then data is read and processed. The post collection processing takes under a second and the Kinect generates the estimates in real time, but the syncing of the two processes is needed.

This effort focused on discrete poses and capturing the threatening events or indicators at certain points. A possible alternative is to analyze a pattern of motion. The patterns could be as short as the motions of drawing a pistol up a few inches out of its holster or the full swing of the arm from drawing to pointing a weapon. The motion aspect could even be combined with the static pose analysis to provide a two pronged approach.

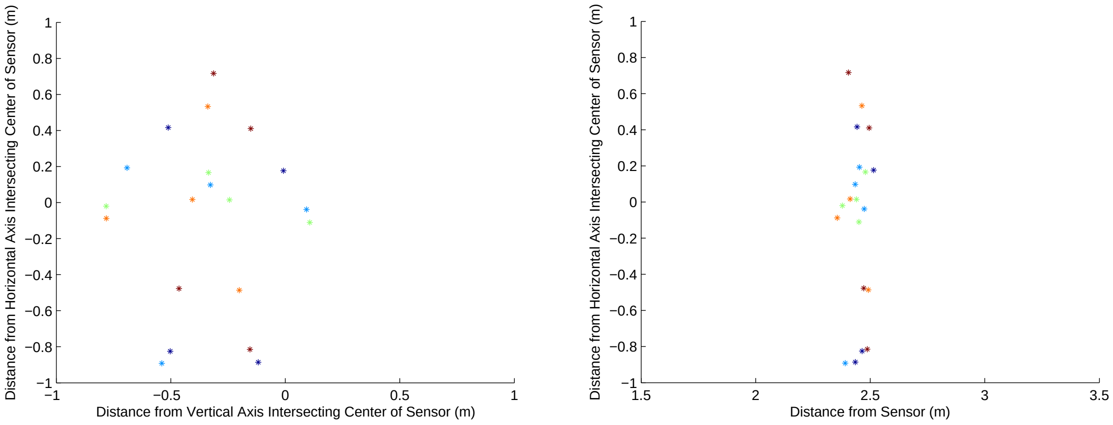
Aside from pose identification, there are many other aspects of a situation that may prove useful to threat identification. The inclusion of context analysis to this method could help to reduce false positives. The current method indicates a threat if the dismounts joints match a trained threat. The ability to determine whether the weapon being held is a toy or an actual weapon would be very useful. Object recognition is just one of many possible additions to a greater threat recognition toolbox approach.

5.3 Contributions

The area of dismount threat recognition has a large number of possible avenues to explore to achieve its goal. This thesis showed the viability of using automatic pose recognition to discern the threat a dismount presents. It also demonstrated the potential of associative memory neural networks which can readily be applied in other ways. If the methods described in this thesis are employed, as part of a larger dismount threat recognition system or alone, they will provide invaluable information to prevent loss of life.

Appendix A. Samples of Collected Poses

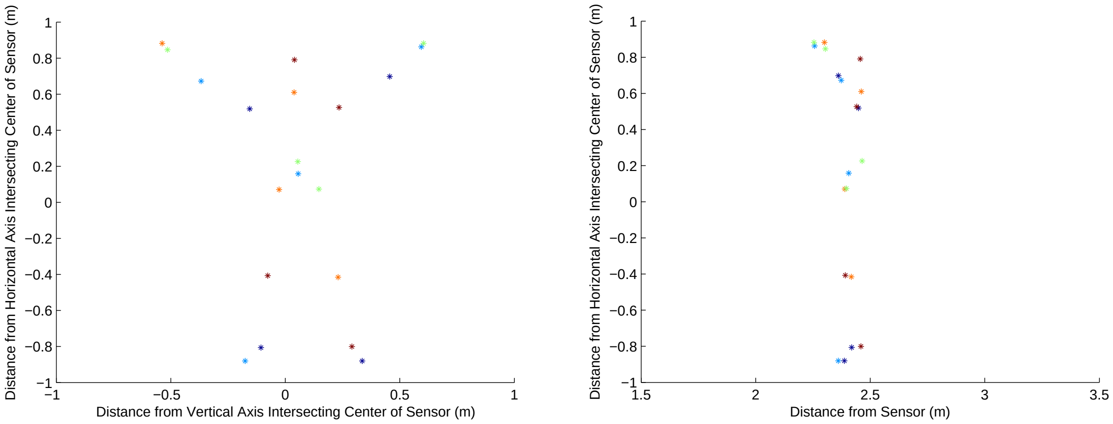
This appendix contains a sample of each pose used for training and testing purposes. The poses are shown in the same order as they are listed in Table 4.1, with a front and side view as they appear in Matrix Laboratory (MatLab).



(a) Front View

(b) Side View

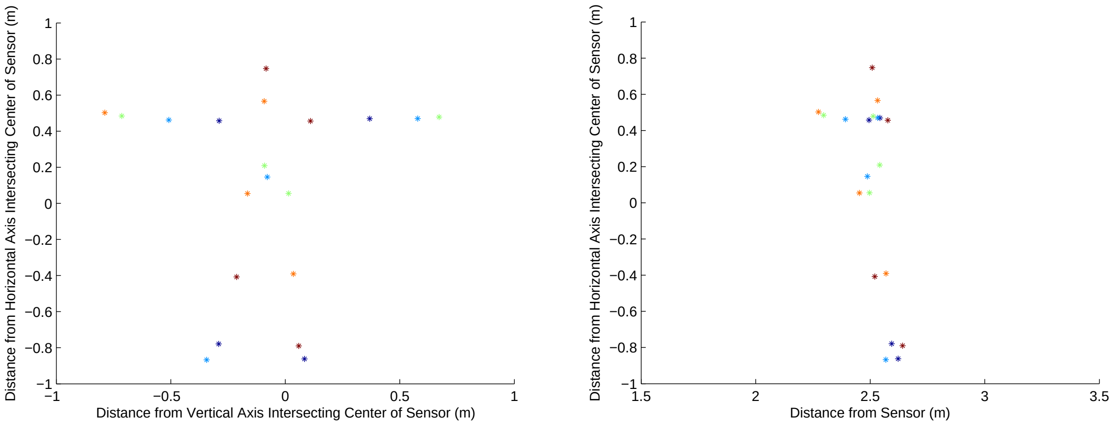
Figure A.1: Pose 1, Standing, arms down and held slightly away from sides, non-threatening.



(a) Front View

(b) Side View

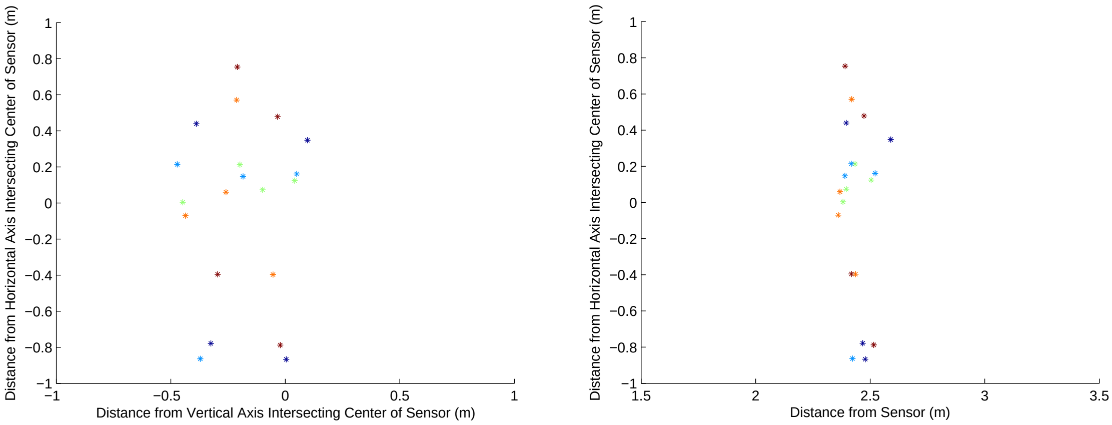
Figure A.2: Pose 2, Standing, arms straight up past the head, non-threatening.



(a) Front View

(b) Side View

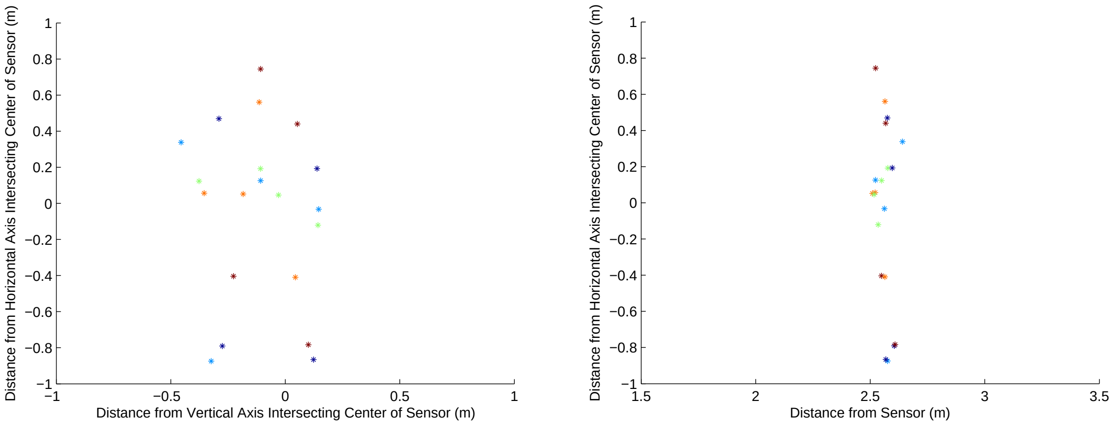
Figure A.3: Pose 3, Standing, arms straight out to the side, non-threatening.



(a) Front View

(b) Side View

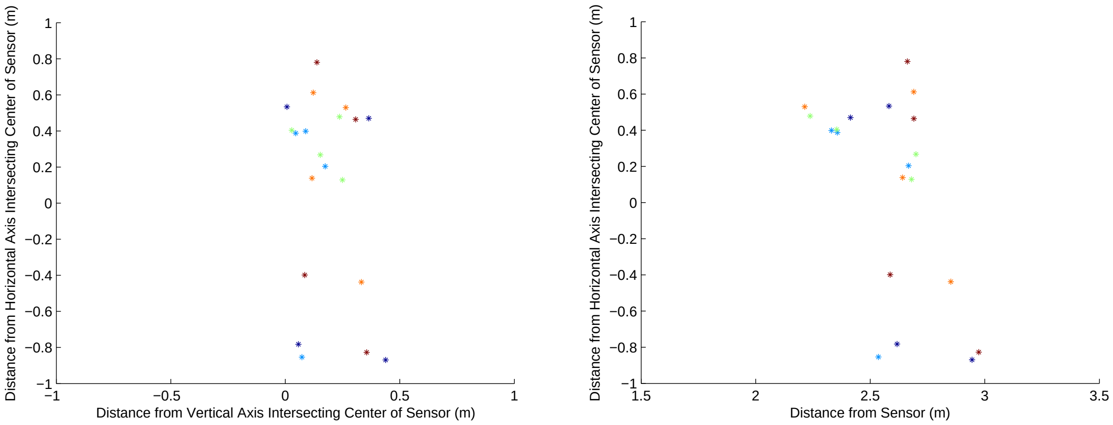
Figure A.4: Pose 4, Standing, drawing a pistol, waist holster, dominant right hand, threatening.



(a) Front View

(b) Side View

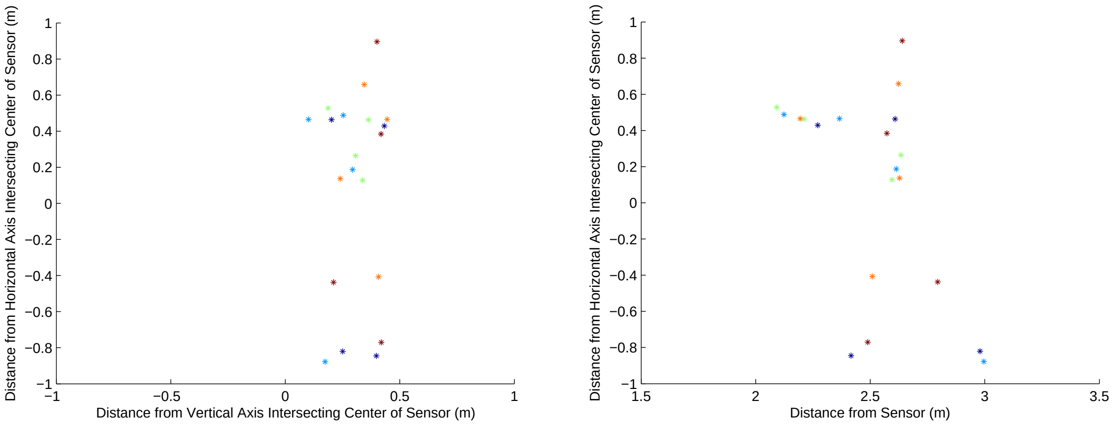
Figure A.5: Pose 5, Standing, drawing a pistol, waist holster, dominant left hand, threatening.



(a) Front View

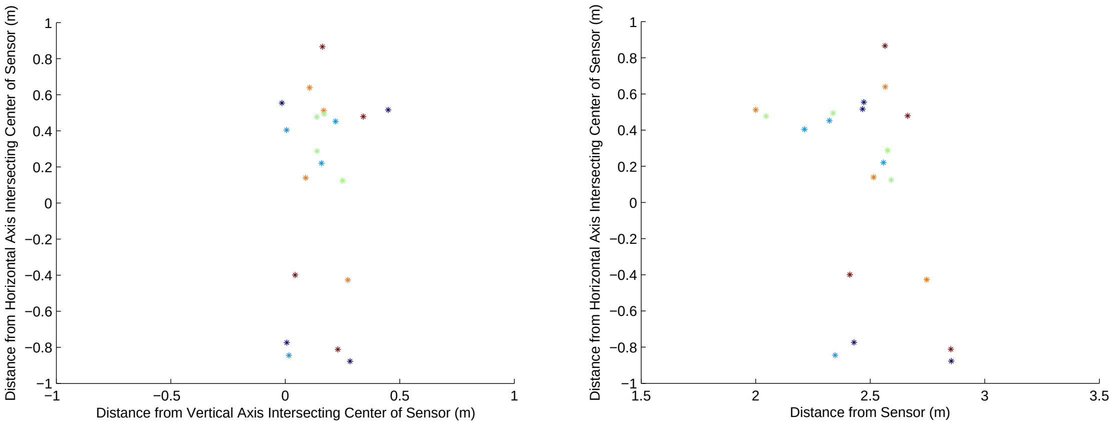
(b) Side View

Figure A.6: Pose 6, Standing, aiming a pistol, dominant right hand, threatening.



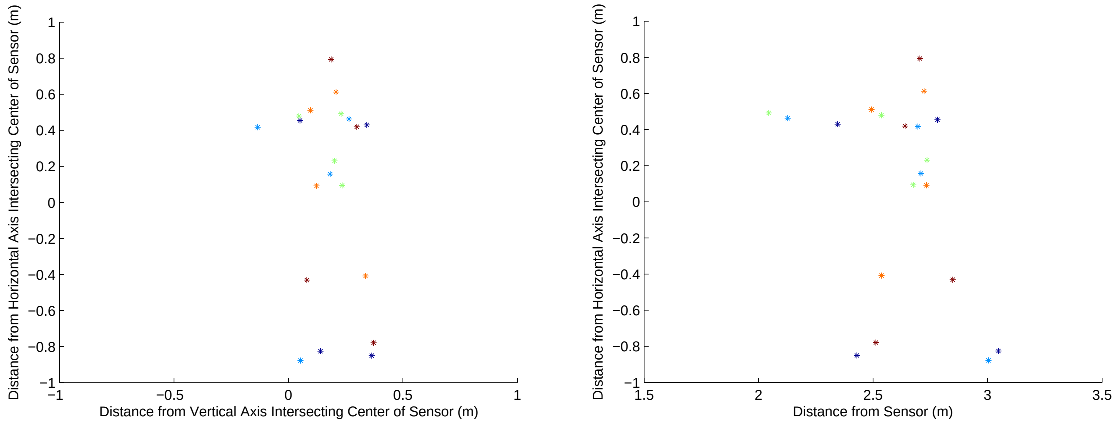
(a) Front View (b) Side View

Figure A.7: Pose 7, Standing, aiming a pistol, dominant left hand, threatening.



(a) Front View (b) Side View

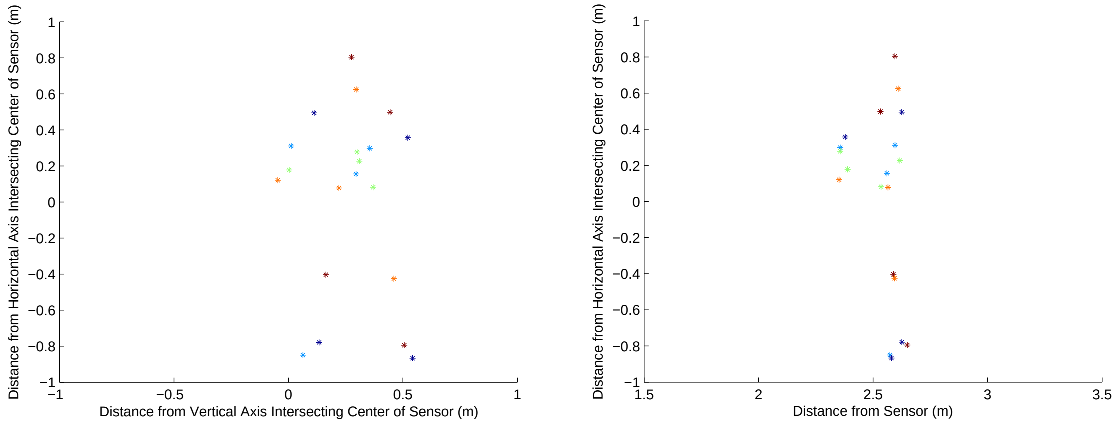
Figure A.8: Pose 8, Standing, aiming a rifle, dominant right hand, threatening.



(a) Front View

(b) Side View

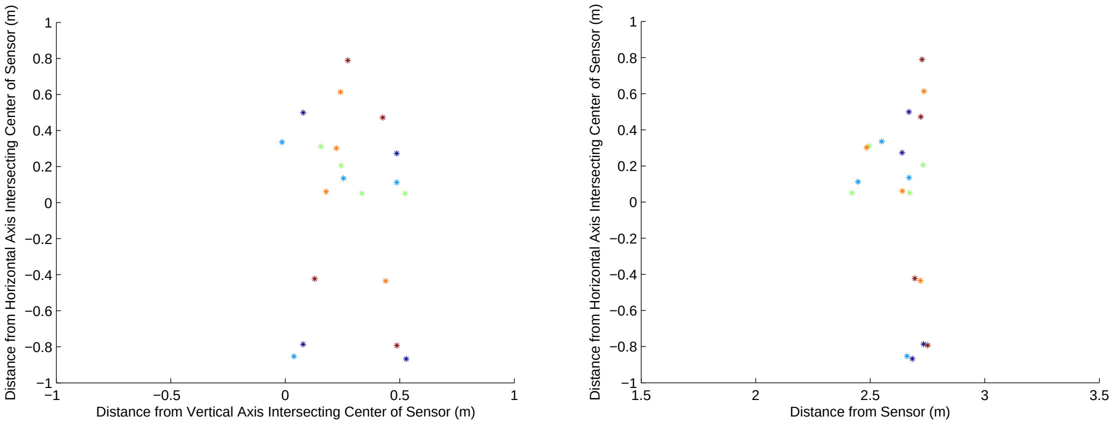
Figure A.9: Pose 9, Standing, aiming a rifle, dominant left hand, threatening.



(a) Front View

(b) Side View

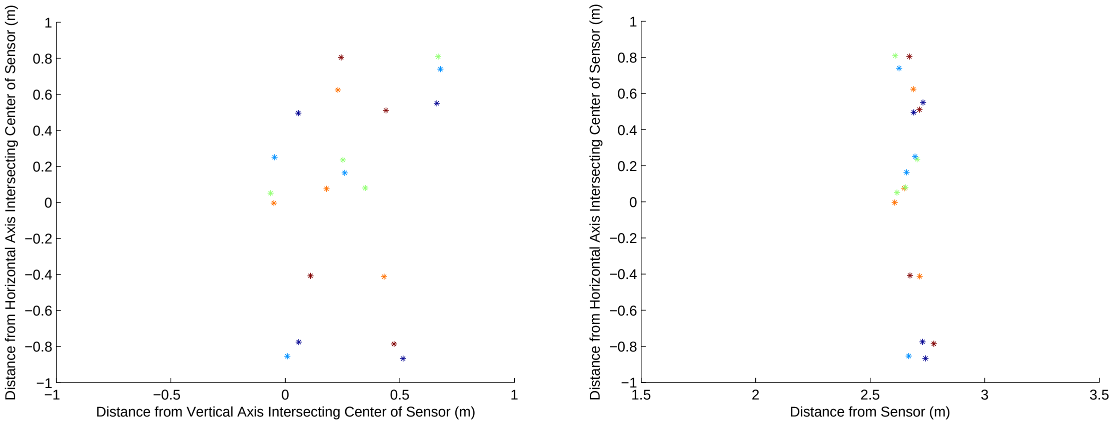
Figure A.10: Pose 10, Standing, carrying a rifle in front of body, dominant right hand, non-threatening.



(a) Front View

(b) Side View

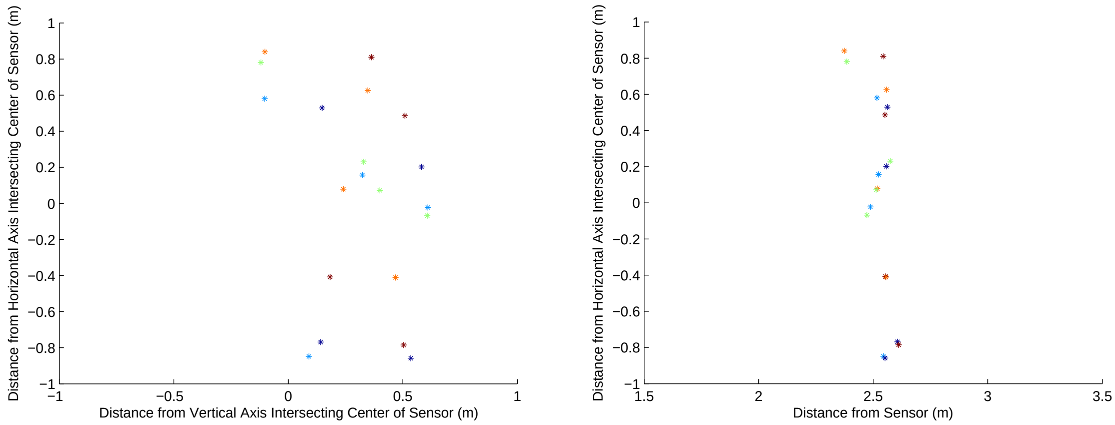
Figure A.11: Pose 11, Standing, carrying a rifle in front of body, dominant left hand, non-threatening.



(a) Front View

(b) Side View

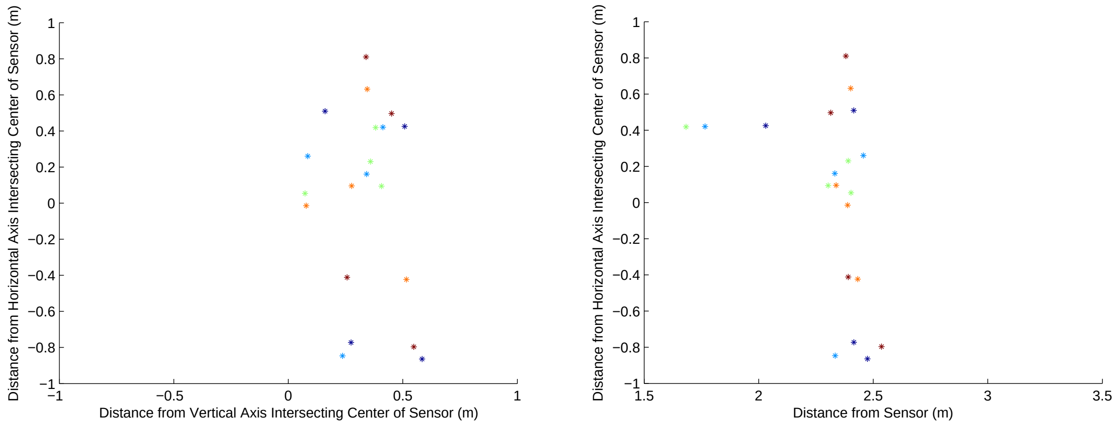
Figure A.12: Pose 12, Standing, waving with right hand in air, left arm at side, non-threatening.



(a) Front View

(b) Side View

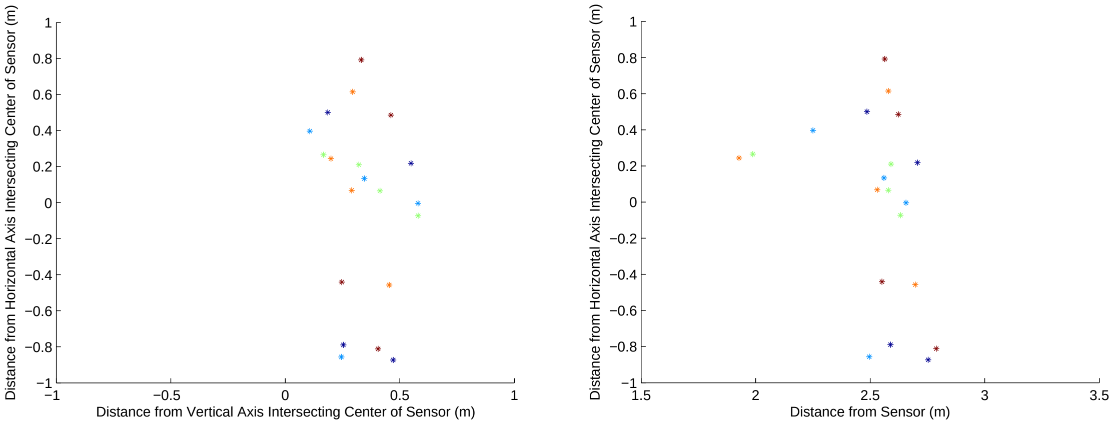
Figure A.13: Pose 13, Standing, waving with left hand in air, right arm at side, non-threatening.



(a) Front View

(b) Side View

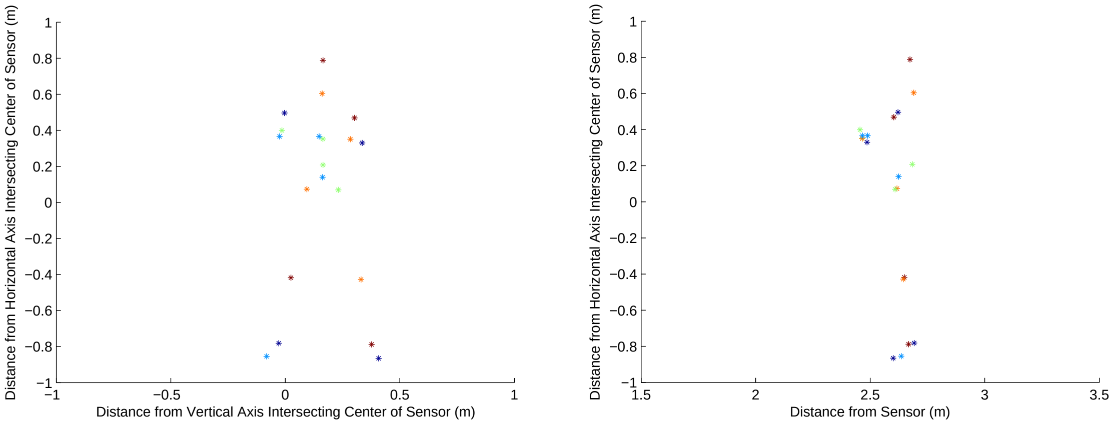
Figure A.14: Pose 14, Standing, reaching for handshake with right arm, non-threatening.



(a) Front View

(b) Side View

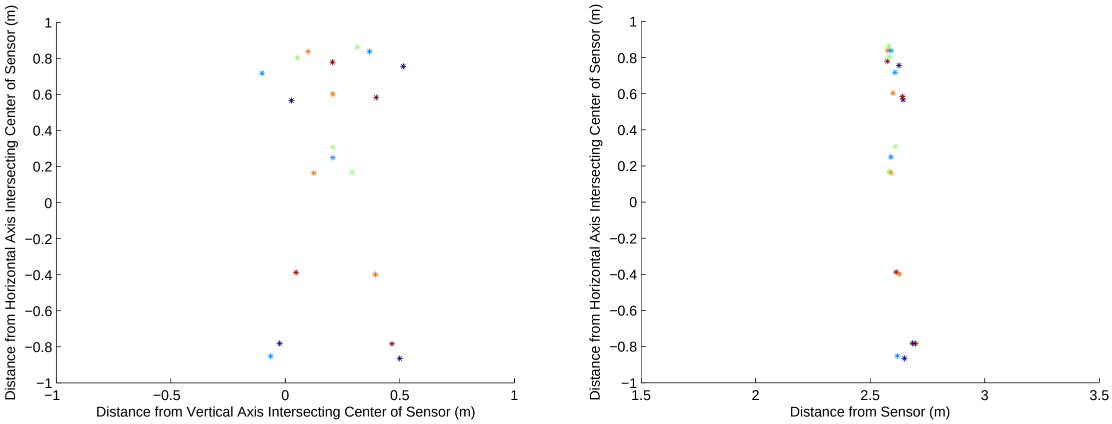
Figure A.15: Pose 15, Standing, reaching for handshake with left arm, non-threatening.



(a) Front View

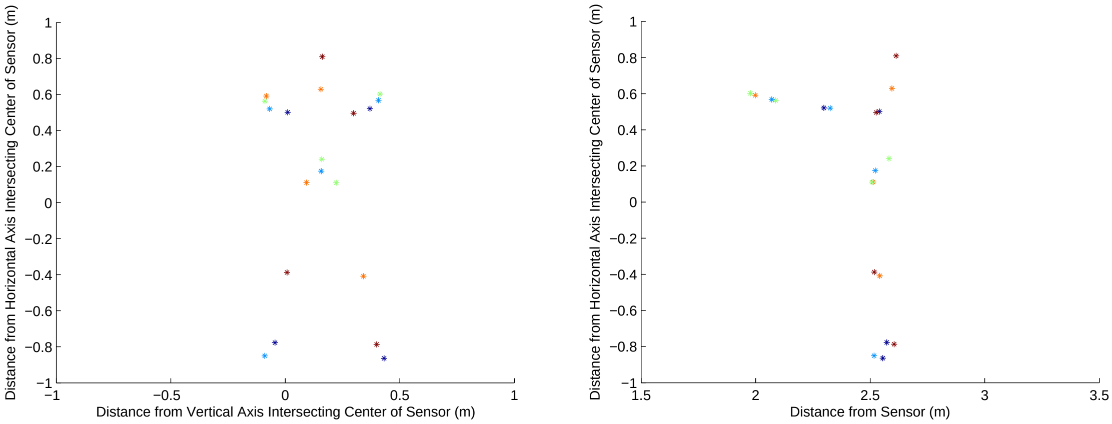
(b) Side View

Figure A.16: Pose 16, Standing, arms crossed in front of chest, non-threatening.



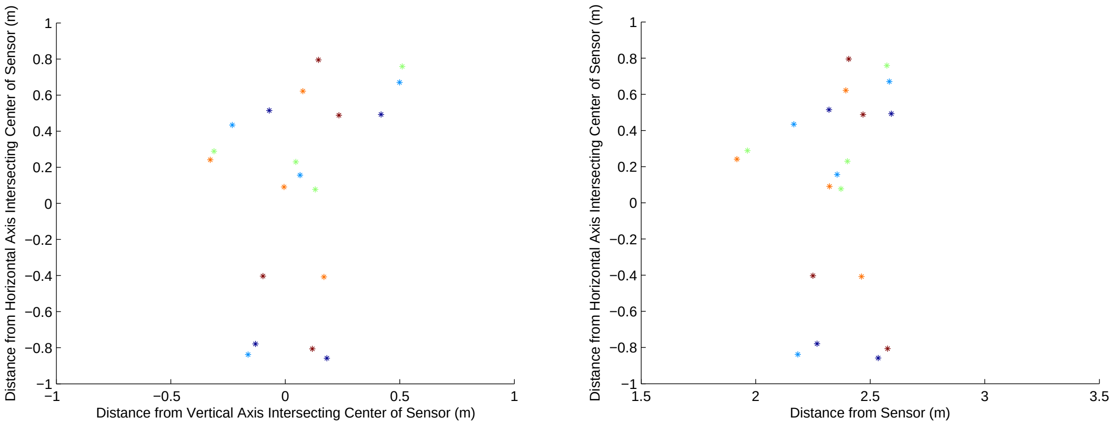
(a) Front View (b) Side View

Figure A.17: Pose 17, Standing, hands behind head in surrender, non-threatening.



(a) Front View (b) Side View

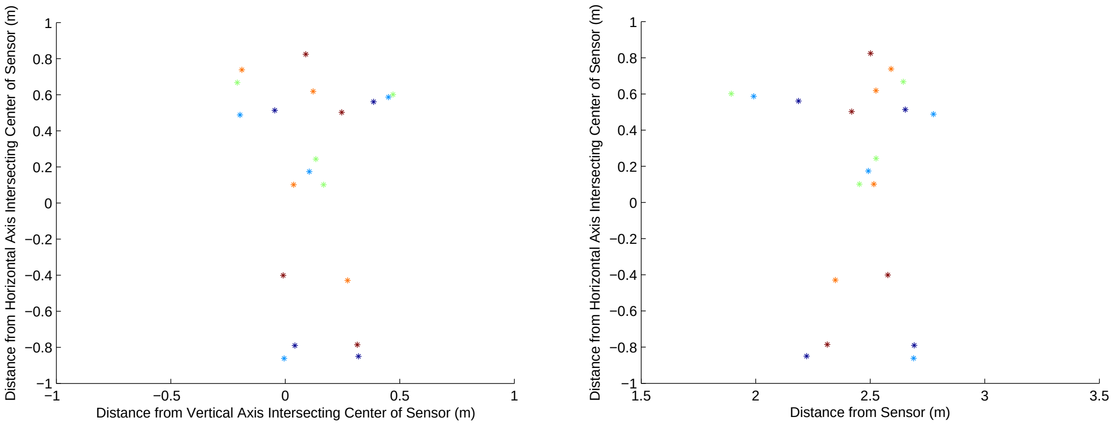
Figure A.18: Pose 18, Standing, arms straight out in front of themselves, non-threatening.



(a) Front View

(b) Side View

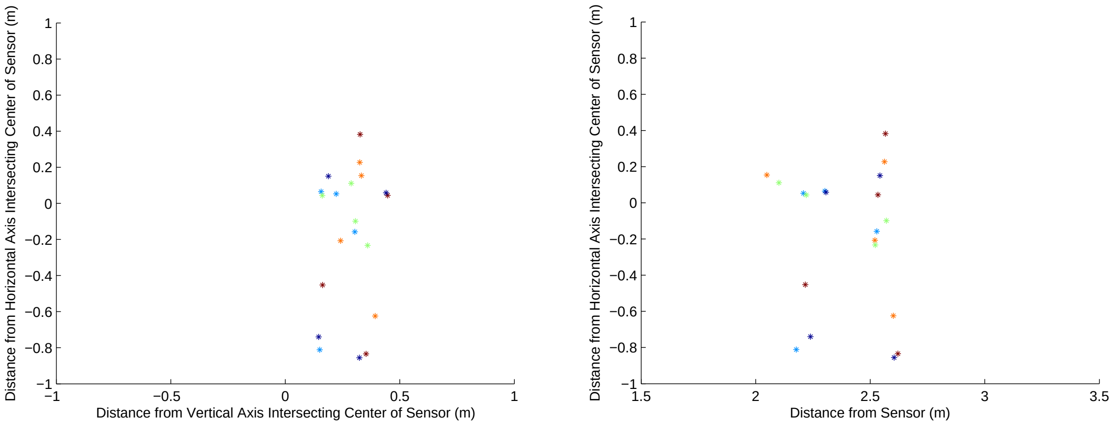
Figure A.19: Pose 19, Standing, throwing grenade with right hand, threatening.



(a) Front View

(b) Side View

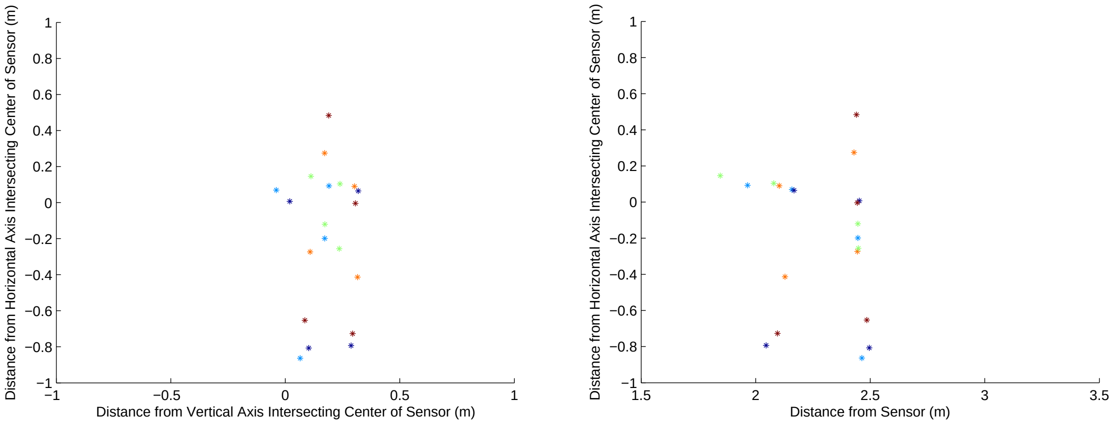
Figure A.20: Pose 20, Standing, throwing grenade with left hand, threatening.



(a) Front View

(b) Side View

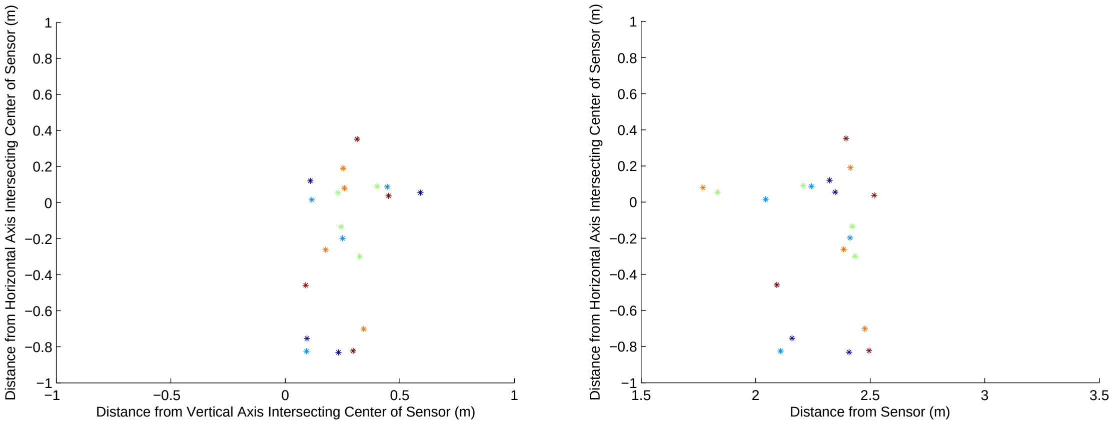
Figure A.21: Pose 21, Kneeling, aiming a pistol, dominant right hand, threatening.



(a) Front View

(b) Side View

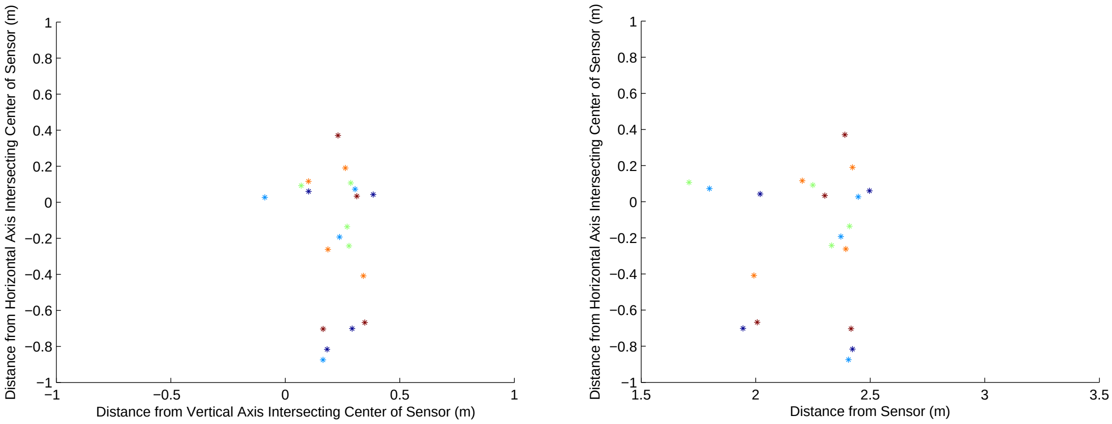
Figure A.22: Pose 22, Kneeling, aiming a pistol, dominant left hand, threatening.



(a) Front View

(b) Side View

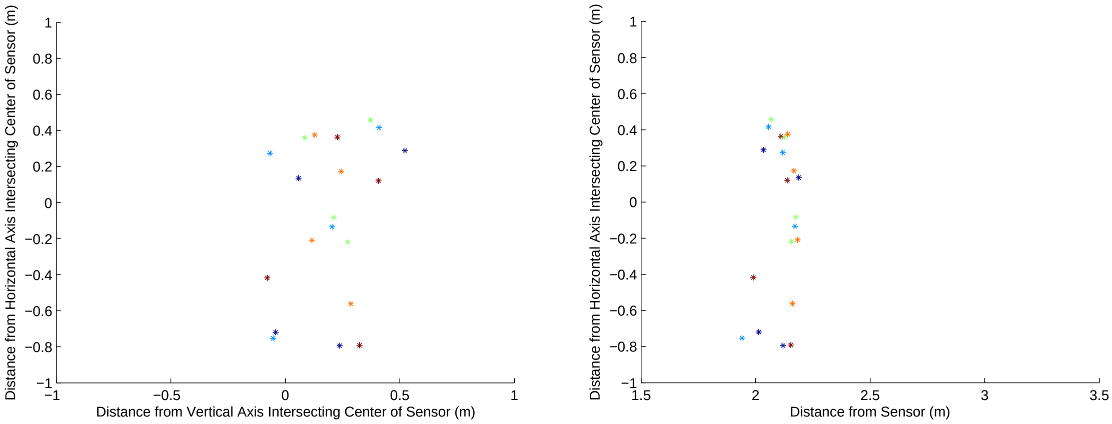
Figure A.23: Pose 23, Kneeling, aiming a rifle, dominant right hand, threatening.



(a) Front View

(b) Side View

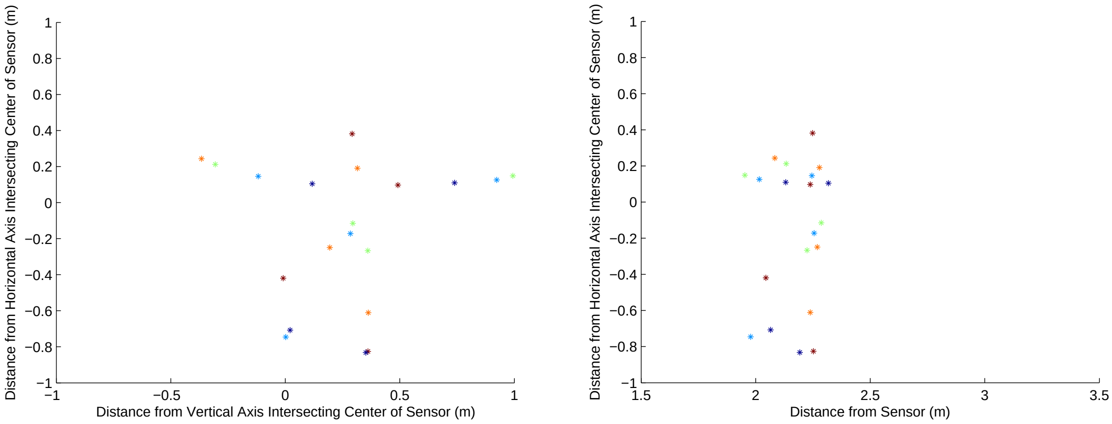
Figure A.24: Pose 24, Kneeling, aiming a rifle, dominant left hand, threatening.



(a) Front View

(b) Side View

Figure A.25: Pose 25, Kneeling, hands behind head in surrender, non-threatening.



(a) Front View

(b) Side View

Figure A.26: Pose 26, Kneeling, arms open as if to hug a child, non-threatening.

Bibliography

1. Draganjac, I., Kovacic, Z., Ujlaki, D., and Mikulic, J., "Dual camera surveillance system for control and alarm generation in security applications," *Industrial Electronics, 2008. ISIE 2008. IEEE International Symposium on*, July 2008, pp. 1070 –1075.

2. Clavel, C., Devillers, L., Richard, G., Vasilexcu, I., and Ehrette, T., "Detection and Analysis of Abnormal Situations Through Fear-Type Acoustic Manifestations," *IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 4, April 2007, pp. IV–21 – IV–24.

3. *Kinect for Xbox 360*, Microsoft Corp., Redmond WA.

4. O'Brien, J. F., Bodenheimer, R. E., Brostow, G. J., and Hodgins, J. K., "Automatic Joint Parameter Estimation from Magnetic Motion Capture Data," *Proceedings of Graphics Interface 2000*, May 2000, pp. 53–60.

5. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A., "Real Time Human Pose Recognition in Parts from Single Depth Images," *Microsoft Research Cambridge & Xbox Incubation*, March 2011.

6. "New USAF Leaders Lay Out Top Priorities," *Defense News*, August 2008.

7. "Northrop Grumman Successfully Demonstrates VADER Dismount Detection," *Global Security*, February 2010.

8. Brooks, A., "Improved multispectral skin detection and its application to search space reduction for dismount detection based on histograms of oriented gradients," Master's thesis, Air Force Institute of Technology, 2010.

9. Clark, J., "Distributed Spacing Stochastic Feature Selection and its Application to Textile Classification," Ph.d. dissertation, Air Force Institute of Technology, 2011.

10. Climer, J., "Overcoming Pose Limitations of a Skin-Cued Histograms of Oriented Gradients Dismount Detector through Contextual Use of Skin Islands and Multiple Support Vector Machines," Master's thesis, Air Force Institute of Technology, 2011.

11. Koch, B., "A Multispectral Bidirectional Reflectance Distribution Function Study of Human Skin for Improved Dismount Detection," Master's thesis, Air Force Institute of Technology, 2011.

12. Nunez, A. S., "A physical model of human skin and its application for search and rescue," Ph.d dissertation, Air Force Institute of Technology, 2009.

13. Poskosky, K., "Design of a Monocular Multi-Spectral Skin Detection, Melanin Estimation, and False Alarm Supression System," Master's thesis, Air Force Institute of Technology, 2010.

14. Leibe, B., Seemann, E., and Schiele, B., "Pedestrian detection in crowded scenes," *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, Vol. 1, June 2005, pp. 878 – 885 vol. 1.

15. Shashua, A., Gdalyahu, Y., and Hayun, G., "Pedestrian detection for driving assistance systems: single-frame classification and system level performance," *2004 IEEE Intelligent Vehicles Symposium*, June 2004, pp. 1 – 6.

16. Rehn, K., Bartlett, D., McCord, B., Berlinger, D., Ellis, J., de Solla, K., Wells, J., Polo, C., Needham, B., Moore, T., and Hollar, A., "The Draw and Fire Sequence," *KR Training*, January 1997.

17. Grafulla-Gonzalez, B., Tomsin, M., Lebart, K., and Harvey, A., "Modelling of millimetre-wave personnel scanners for automated detection," *Imaging for Crime Detection and Prevention, 2005. ICDP 2005. The IEE International Symposium on*, June 2005, pp. 9 – 13.

18. Cohen, C., "A control theoretic method for categorizing visual imagery as human motion behaviors," *Applied Imagery and Pattern Recognition Workshop, 2005. Proceedings. 34th*, Dec. 2005, pp. 8 – 191.

19. Cohen, C., Morelli, F., and Scott, K., "A Surveillance System for the Recognition of Intent within Individuals and Crowds," *2008 IEEE Conference on Technologies for Homeland Security*, May 2008, pp. 559 –565.

20. Banczyk, K. and Krawczyk, H., "A model of an ontology oriented threat detection system (OOTDS)," *International Conference on Information Technology*, , No. 1, May 2008, pp. 1–4.

21. Grest, D., Woetzel, J., and Koch, R., "Nonlinear body pose estimation from depth images," *Deutsche Arbeitsgemeinschaft für Mustererkennung Symposium*, 2005.

22. Anguelov, D., Taskar, B., Chatalbashev, V., Koller, D., Gupta, D., Heitz, G., and Ng, A., "Discriminitive learning of markov random fields for segmentation of 3D scan data," *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR, 2005, pp. 169–176.

23. Zhu, Y. and Fujimura, K., "Constrained optimization for human pose estimation from depth sequences," Asian Conference on Computer Vision, 2007.

24. Plagemann, C., Ganapathi, V., Koller, D., and Thrun, S., "Real-time identification and localization of body parts from depth images," *IEEE International Conference on Robotics and Automation*, 2010.

25. Haykin, S., *Neural Networks*, Prentice Hall PTR, 1st ed., 1994.

26. Morell, J., “Using the Xbox Kinect and the MySkeleton Program for Skeleton Extraction,” User guide, Air Force Institute of Technology, August 2011.

27. Mendenhall, M., “Introduction to Artificial Neural Networks,” 2011, Class Notes.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
22-03-2012	Master's Thesis	March 2010 - March 2012

4. TITLE AND SUBTITLE
Dismount Threat Recognition through Automatic Pose Identification

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Freeman, Andrew M., Captain

5d. PROJECT NUMBER
12G408

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
Wright-Patterson AFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GE/ENG/12-14

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratories, Sensors, ATR, Target Recognition Branch
Mrs. Olga Mendoza-Schrock, Research Mathematician
2241 Avionics Circle Area B, Ste 18
Wright Patterson AFB, Ohio 45433-7320
olga.mendoza-schrock@wpafb.af.mil
(937) 798-8591

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RYAT

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution A. Approved for Public Release; Distribution Unlimited.

13. SUPPLEMENTARY NOTES
This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT
Analyzing the actions that precede hostile events yields information about how the event occurred and uncovers warning signs that are useful in the prediction and prevention of future events. A dismounts posturing, or pose, indicate what they are about to do. Pose recognition and identification is a topic of study that can be utilized to discern this threat information. Pose recognition is the process of observing a scene through an imaging device(s), determining that a dismount is present, identifying the three dimensional (3D) position of the dismount's joints, and evaluating what the current configuration of the joints means. This thesis explores the use of automatic pose recognition to identify threatening poses and postures by means of an artificial neural network. The data is collected utilizing the depth camera and joint estimation software of the Kinect® for Xbox360. A threat determination is made based on the pose identified by the network. Accuracy is measured both by the correct identification of the pose presented to the network, and proper threat discernment. A high level of threat determination accuracy is achieved indicating that automatic pose recognition is a promising means of discerning if a dismount is threatening.

15. SUBJECT TERMS
Dismount Threat Recognition, Threat Determination, Pose Identification, Pose Recognition, Associative Memory Neural Networks

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					Jeffrey D. Clark, LtCol
REPORT	**ABSTRACT**	**c. THIS PAGE**	UU	72	**19b. TELEPHONE NUMBER** *(Include area code)*
U	U	U			(937) 255-3636 x4614 jeffrey.clark@afit.edu