

University of Memphis

University of Memphis Digital Commons

Electronic Theses and Dissertations

4-15-2011

Real-Time Object Classification Using a Custom Sparse Array Profile Sensor on an Embedded Microcontroller

Robert Kenneth Reynolds Jr.

Follow this and additional works at: <https://digitalcommons.memphis.edu/etd>

Recommended Citation

Reynolds, Robert Kenneth Jr., "Real-Time Object Classification Using a Custom Sparse Array Profile Sensor on an Embedded Microcontroller" (2011). *Electronic Theses and Dissertations*. 173.
<https://digitalcommons.memphis.edu/etd/173>

This Thesis is brought to you for free and open access by University of Memphis Digital Commons. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of University of Memphis Digital Commons. For more information, please contact khhgerty@memphis.edu.

To the University Council:

The Thesis Committee for Robert Kenneth Reynolds, Jr, certifies that this is the final approved version of the following electronic thesis: “Real-Time Object Classification Using a Custom Sparse Array Profile Sensor on an Embedded Microcontroller.”

Eddie L. Jacobs, D. Sc
Major Professor

We have read this thesis and recommend
its acceptance:

David J. Russomanno, Ph.D.

Srikant Chari, Ph.D.

Accepted for the Graduate Council:

Karen D. Weddle-West, Ph.D.
Vice Provost for Graduate Programs

REAL-TIME OBJECT CLASSIFICATION USING A CUSTOM SPARSE ARRAY
PROFILE SENSOR ON AN EMBEDDED MICROCONTROLLER

by

Robert K. Reynolds, Jr.

A Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

Major: Electrical and Computer Engineering

The University of Memphis

May 2011

Abstract

Reynolds, Robert Kenneth Jr. M.S. The University of Memphis. May 2011. Real-time Object Classification Using a Custom Sparse Array Profile Sensor on an Embedded Microcontroller. Major Professor: Eddie L. Jacobs, D. Sc.

This thesis presents implementation of data acquisition and object classification algorithms on a low-resource microcontroller for real-time, broad-scale object classification using a low-cost sparse detector imaging sensor. The sensor is designed to detect and classify objects into the broad categories of human, vehicle, or animal, making note of objects of high interest. This paper encompasses software for implementation onboard a low-resource microcontroller platform to acquire, process, and classify crude images of subjects for classification purposes. This paper also encompasses improvements made to a prototype hardware system to form a custom sensor array from commercially available, off-the-shelf hardware components. The sensor is designed for deployment scenarios to monitor vast geographic areas where broad monitoring is required with low false-alarm rates generated by objects of less interest.

Table of Contents

| Chapter | | Page |
|---------|--|------|
| | List of Tables | iv |
| | List of Figures | v |
| | Key to Abbreviations | vii |
| 1 | Introduction | 1 |
| 1.1 | Profiling Sensors | 1 |
| 1.2 | Review of prior efforts | 5 |
| 1.3 | Preview of Thesis | 9 |
| 1.4 | Thesis Statement | 9 |
| 2 | Hardware Improvements | 10 |
| 2.1 | PVC Profile Sensor | 10 |
| 3 | Microcontroller coupled with vertically oriented detector elements | 12 |
| 3.1 | Microcontroller Hardware | 12 |
| 3.2 | Feature Extraction and Classification | 17 |
| 3.3 | Classification Results | 21 |
| 4 | Custom Sensor Array | 22 |
| 4.1 | Introduction | 22 |
| 4.2 | Hardware Configuration | 23 |
| 4.3 | Simple classification technique using custom array | 25 |
| 4.4 | Custom array profile reconstruction technique | 27 |
| 4.4.1 | Implementation of Profile Reconstruction | 31 |
| 4.4.2 | Velocity calculation and incorporation | 34 |
| 4.5 | Results | 38 |
| 5 | Conclusions and Future Work | 50 |
| 5.1 | Summary | 50 |
| 5.2 | Works in progress / Future work: | 51 |
| 6 | References | 55 |
| | APPENDIX | 57 |

List of Tables

| Table List | Table Title | Page |
|------------|---|------|
| Table 4-1 | Real-time Row Energy Classification Confusion Matrix | 41 |
| Table 4-2 | Example Human profiles, Real-Time Reconstruction | 42 |
| Table 4-3 | Example Animal Profiles, Real-Time Reconstruction | 42 |
| Table 4-4 | Example Vehicle Profiles, Real-Time Reconstruction | 42 |
| Table 4-5 | Post-Processed profiles demonstrating poor rebuilding | 45 |
| Table 4-6 | Height/Width Feature Confusion Matrix | 46 |
| Table 4-7 | Six-Feature Classification Confusion Matrix | 49 |
| Table 4-8 | Classification rates for profile features | 49 |

List of Figures

| Figure List | Figure Title | Page |
|-------------|---|------|
| Figure 1-1 | Dog and human passing through the parallel detector beams of a sparse sensor array. | 3 |
| Figure 1-2 | Example profile from a sparse sensor array. | 4 |
| Figure 1-3 | CX-RVM5 Near-IR retroreflective detector. | 6 |
| Figure 1-4 | A prototype 128 element pyroelectric array designed by NVESD, Dept. of the Army [10]. | 8 |
| Figure 2-1 | Paired detectors mounted within PVC housings. | 10 |
| Figure 3-1 | Digi® BL4S200 Single Board Computer [17]. | 13 |
| Figure 3-2 | Height/Width Ratio feature for samples in the Profile Sensor 'Timing' library. | 18 |
| Figure 3-3 | Human, animal and vehicle profiles and derived feature spaces. | 19 |
| Figure 4-1 | Prototype custom array utilizing a 4'x8' pegboard. | 23 |
| Figure 4-2 | Example of custom array implementation using PVC pipe. | 24 |
| Figure 4-3 | Normalized energy along rows for samples of the profiling sensor library. | 27 |
| Figure 4-4 | Subject passing through a vertical column sensor. | 28 |
| Figure 4-5 | Straight vertical object passing through vertical array. | 28 |
| Figure 4-6 | Straight vertical object passing through custom array. | 29 |
| Figure 4-7 | Subject profile reconstruction after passing through broken array. | 30 |
| Figure 4-8 | Custom profile sensor array prototype realized on PVC pipe. | 31 |
| Figure 4-9 | Person carrying a vertical object through the custom sensor array of Figure 4-8; (a) uncorrected and (b) corrected. Vertical beam is shown in oval. | 32 |

| | | |
|-------------|---|----|
| Figure 4-10 | Corrected data sample collected through custom array. | 32 |
| Figure 4-11 | Incorrect 'Corrected' profiles. | 34 |
| Figure 4-12 | Calculated detector positions across multiple calibration passes. | 35 |
| Figure 4-13 | More accurately reconstructed profiles of Figure 4-11 generated by velocity incorporation. | 37 |
| Figure 4-14 | Histogram of object velocities from a data collection event. | 39 |
| Figure 4-15 | Raw data acquired from custom array (a) and Reconstructed Profile (b). | 40 |
| Figure 4-16 | Raw (a) and rebuilt (b) profile of a dog illustrating noise introduction (circled in red). | 43 |
| Figure 4-17 | Rebuilt profile of Figure 4-16 without noise. | 44 |
| Figure 4-18 | Six-Feature classification rate % dependent upon box width and divider placement from center. | 47 |
| Figure 4-19 | Six-Feature classification rates for box width 500, demonstrating peak classification rate. | 48 |
| Figure 5-1 | Custom Array with parallel beams and multiple reflector pole pairings, viewed from overhead. | 52 |
| Figure 5-2 | Custom Array with Single Reflector Pole. | 53 |

Key to Abbreviations

COTS – Commercially available Off-The-Shelf

FOV – Field of View

GIS – Geographic Information Service

LWIR – Long Wave Infrared

NVESD - Night Vision and Electronic Sensors Directorate

OTW – Optical Trip Wire

PVC – Poly-Vinyl Chloride (plastic)

SBC – Single Board Computer

SBI – Secure Border Initiative

SSA – Sparse Sensor Array

UGS – Unattended Ground Sensor

USB – Universal Serial Bus

1 Introduction

1.1 Profiling Sensors

Persistent security monitoring is required to protect areas from intruders. Resources are often limited for persistent monitoring, especially when the area of patrol covers vast geographical regions that are largely uninhabited. Example scenarios include, but are not limited to, the protection of military installations and border monitoring. Border monitoring along the US-Mexican border is of particular interest, where large quantities of illegal drugs are often smuggled on foot into the United States along drug routes. The vast, rugged terrain of the US Southwest is too large to monitor with limited manpower resources. Deployment of Unattended Ground Systems (UGS) is a potential solution to monitoring, but is expensive due to the relatively high cost of detector systems that they employ. Additionally, UGS systems are often unable to distinguish between humans and animals that occur naturally in the deployment environment, resulting in a large number of false alarm detections.

A relatively low-cost profiling sensor system was conceptualized by Ronnie Sartain [1] to improve the number of false alarm detections. Requirements for the sensor system include low cost, minimal power consumption, low bandwidth, easy deployment and concealment, and the ability to reduce false alarm detections by classifying objects automatically. Objects are classified into three broad categories as either human, vehicle, or animal. The sensor operates by gathering a crude image of an object, referred to as a silhouette, as it passes through the sensor. Classifications are performed based upon the shape of the silhouette. Objects classified as animals by the sensor are likely natural inhabitants of the deployment area and pose no cause for alarm. Human and vehicle

classifications, though, likely have no business in the remote area and are worthy of further review by either other sensor systems or human patrol dispatching.

Concealment is an important requirement in the design of the profile sensor. If the sensor is discovered, it can be defeated or destroyed. Drug smuggling teams have been known to look particularly for sensors along frequently used trails. The system must also be able to operate for long periods of time with little or no maintenance. Routine maintenance to the sensor could easily expose its location and possibly counter its concealment. With little to no maintenance to replace batteries, power requirements must therefore be minimal. Manpower required to replace batteries on a routine basis can become expensive, especially if a large number of sensors are deployed.

Low bandwidth communications are also a requirement for the system. Sending large amounts of data requires that a transmitting radio utilize a larger bandwidth, thereby requiring more power to make the transmission possible. Detection events of naturally dwelling animals need not be reported, further reducing power requirements. It should be noted that the sensor need only indicate unusual detection events. Drug cartels employing smugglers have become quite sophisticated lately in counter-detection techniques. Cartels have been known to employ electronic 'sniffer' teams to detect and counter surveillance systems. A high-bandwidth radio is more likely to be detected and countered than a low bandwidth radio.

Cost is another major requirement for the sensor. The SBI.net project under development for border security by the Department of Homeland Security has an estimated price tag of nearly \$1 million dollars per mile of border [2]. Monitoring the entire 2000 mile border between the US and Mexico with such a system places an

enormous burden on taxpayers. A more cost effective system is therefore warranted. The profile sensor system is defined to be a low cost system in means of construction, deployment, maintenance, and replacement.

Sartain's profile sensor operates by employing a series of optical trip wire (OTW) detectors along a vertical column, spaced approximately 5 inches apart with the beams facing the same direction in a parallel configuration as illustrated in Figure 1-1. Active retro-reflector detectors are employed in a staring configuration, each sending out a signal to an individually mated reflector. The relatively short range of the retro-reflective elements (approximately five meters) makes the profile sensor particularly suited for narrow pathways and bottleneck areas created by natural landforms.



Figure 1-1. Dog and human passing through the parallel detector beams of a sparse sensor array.

This configuration forms a type of electronic fence, commonly referred to as a sparse sensor array. Objects are 'scanned' as they pass through the array of detectors.

The OTW detectors work in a strictly digital format; a beam is either continuous or broken, returning an on or off state, respectively. By making note of exactly when an individual detector is tripped and reset, the amount of time that an object spent within the gates can be calculated. Timing between all the detectors is synchronized by their placement along the same plane. Thus, when the time series of the states of all the detectors are shown together, a crude image of the object is shown as it passed through the beams of the detector. This crude image, referred to as a silhouette or profile, is unique to the object and can be used to classify the object. An example silhouette of a horse is displayed in Figure 1-2.

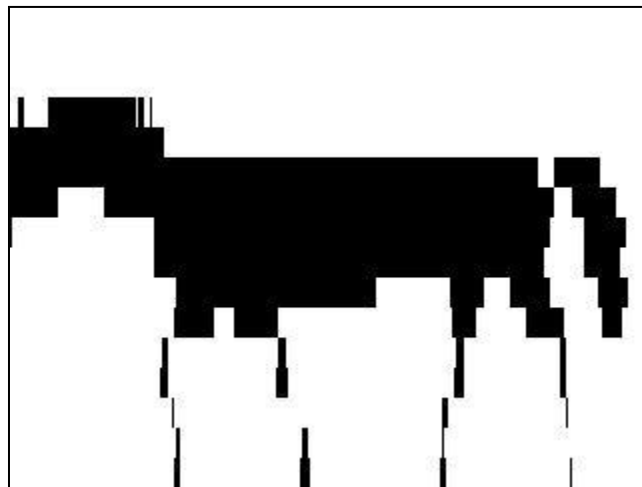


Figure 1-2. Example profile from a sparse sensor array.

The sparse sensor array has advantages over traditional imaging sensors in the fact that it can be used to monitor areas over great distances. The divergent field-of-view of a traditional imaging sensor, such as a camera, is unable to distinguish between small objects near the sensor and large objects that are far away. Both objects fill the field of view of the sensor, but without knowing the distance from the sensor, calculating the

object's exact height is impossible. The sparse sensor array, though, can determine the height of the passing object independently of the distance from the object to the sensor array. However, detector elements must be placed at least the same height as the passing objects it is to detect. This makes the sensor array quite tall and makes it more difficult to hide.

Silhouettes are classified by observing features and making a comparison to silhouettes of known classifications. One of the easiest features to extract from a silhouette is performed by noting the ratio of its height to its width. It was noticed that the profiles of humans, walking upright, have a very high height-to-width ratio as compared to those of vehicles and animals. More robust feature extraction techniques have also been implemented to make a reliable classification. However, these additional feature extraction and classification techniques require additional computational resources.

This paper encompasses hardware improvements made to the detector array to aid in both deployment as well as concealment. Additionally, this paper discusses implementation of a microcontroller to form a standalone sensor system capable of acquiring and classifying a profile independently of human intervention.

1.2 Review of prior efforts

Requirements for a robust UGS with low false-detection are discussed by Sartain [1]. The sensor must be able to properly classify passing objects as either human, vehicle, or animal. The sensor must meet requirements of UGS systems fielded already, such as having relatively low-cost, small, low power consumption, lightweight, and covert deployment. A prototype sparse sensor array was realized by Russomanno *et al*

[3] by implementing 16 commercially-off-the-shelf (COTS) near-infrared, retro-reflective detectors for a proof of concept study. Each of the CX-RVM5 active near-infrared detectors manufactured by SunX corporation is mated to their own individual reflector; maximum distance separation between the detector to its reflector is 5 meters [4]. The status of the OTW beams is either broken or unbroken, creating an off or on signal, respectively. These signals are recorded by computer through the means of a USB data acquisition system [5] to form a complete profiling sensor.

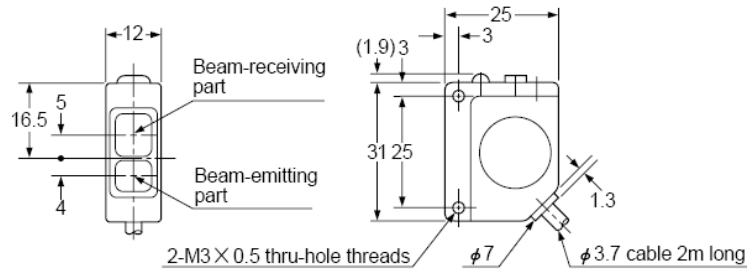


Figure 1-3. CX-RVM5 Near-IR retroreflective detector [2].

Data recorded by the sensor is two dimensional with respect to a height-to-width observation. The width of the profile is a measure of the amount of time that an object breaks the beams of the array. This time is a function of the object's physical length and the velocity that it travels. Assuming that most objects travel at a normal walking speed, the profile 'width' becomes a representation of the object's length as it passes through the sensor.

Classification of an object to the broad categories of human, vehicle, or animal is performed by monitoring various features of the profiles. Classification is typically performed though post-processing algorithms. High classification rates were reported by Russomanno *et al.* [2] for the broad categories of human vs. non-human on initial data collected. Yeasin *et al.* [6] was able to further identify additional features on profiles by

means of various machine learning algorithms. Humans with no packs/small packs, humans with large packs, and miscellaneous objects were correctly classified 83% to 91% for individual classification algorithms.

Prior data collection events with the prototype profile sensor array have created an extensive library of animal, vehicle, and animal profiles [7]. Additional field collections have expanded this library to over 1000 profiles. Profile lengths of this library are based solely on the amount of time that the object remains in the field of view of the sensor. Data collection events revealed that the profile generated by an object depends on the speed at which it moves through the sensor's field of view as well as the object's physical length. As a result, the profile length of fast moving large object could potentially be the same length as a smaller, slowly moving object. Chari *et al.* [8] discuss a technique in which velocity can be estimated for an object, thereby calculating a more accurate width of the object, improving classification rates to as high as 99%.

Klett *et al.* propose an alternative approach to a profiling sensor by placing the entire detector array behind a single optical system [9]. This approach greatly reduces the amount of hardware required to acquire a profile, aiding in concealment. A variety of passive technologies are available to acquire data on a passing subject, including focal plane arrays and linear arrays, pyroelectric and microbolometer detectors. A 128-element pyroelectric array has been prototyped by the US Army Night Vision Electronic Sensors Directorate (NVESD) and is illustrated in Figure 1-4 [10]. Data collected with this type of sensor have led to significant advancements in the development of classification algorithms.

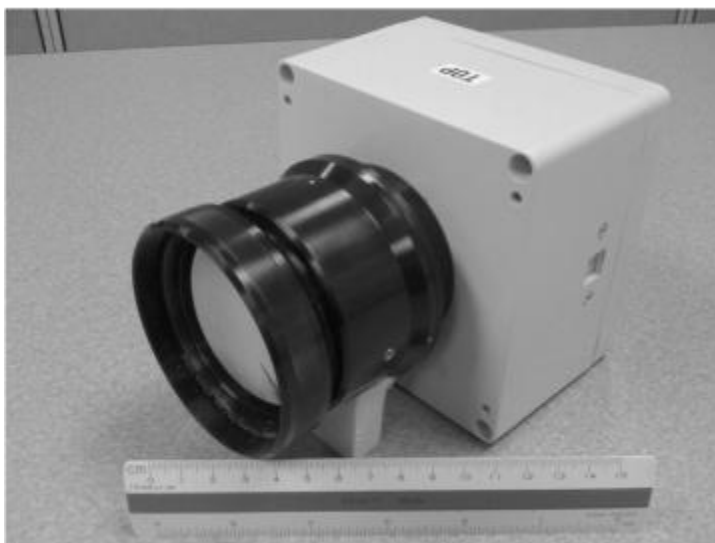


Figure 1-4. A prototype 128 element pyroelectric array designed by NVESD, Dept. of the Army [10].

Passive systems such as the linear array and conventional imaging systems are advantageous in that their hardware is typically much smaller than the OTW system. However, the high resolution data generated by the linear array and conventional imaging system must be converted to a binary image for useful data extraction, requiring additional computational resources. Russomanno *et al* [11] proposed a method of concealment in which the detectors of a sparse sensor array are distributed horizontally along the path of travel.

A primary feature used in classifying a particular profile is the profile's height to width ratio [1, 10]. Estimation of an object's velocity can be used to make a more accurate width calculation, thereby yielding a better height-to-width ratio and increased likelihood of proper classification [8]. Classification rates as high as 99% have been achieved by the incorporation of velocity estimation on data collected by a long-wave infrared (LWIR) camera.

Additional work includes the development web-service interface tools designed to interface to a network of independent, autonomous sensors [12]. Sensor systems such as the sparse sensor array can be tasked to provide notification that an object was detected across the network-centric environment. Geographical Information System (GIS) has been utilized in the development of software to monitor, locate, task, and retrieve information from multiple sensors across geographical areas [13].

1.3 Preview of Thesis

This thesis presents three contributions towards continuing development of the sparse sensor detector:

1. Improvement of prototype hardware.
2. Implementation and analysis of classification algorithms onto a low-resource microcontroller for real-time classification.
3. Implementation of a reconfigurable prototype custom sensor array.

Major hardware enhancements and algorithms presented within this thesis expand the framework for continuing research with the sparse sensor array.

1.4 Thesis Statement

A relatively low-cost microcontroller can be utilized for real-time broad-scale object classification for custom optical trip wire profiling sensor systems.

2 Hardware Improvements

2.1 PVC Profile Sensor

Deployment of the prototype conventional profile sensor is hindered by its large size. Additionally, individual detector elements, although durable, do require some type of protection from potential damage caused by shipment to deployment areas. Both these problems were alleviated with the construction of a profile sensor array made of PVC pipe designed with deployment and scalability in mind. Detectors of this array are housed within a T-section of 1-1/2" pipe coupling with the detector beam direction passing through the perpendicular axis. Housing also protects the sensor from ambient light shining in; sunlight is somewhat blocked. If necessary, 1-1/2" tubes can be added for additional shading. Two T-couplings with their internal detectors are paired into two-element module with detectors 5" apart from each other, similar to the profile sensor prototype.



Figure 2-1. Paired detectors mounted within PVC housings.

Eight detector pairs are stacked at 5” intervals to make a profile sensor virtually equivalent to the prototype discussed earlier. Detector pairing modularity also lends itself to quick replacement; if a single detector is found to be defective, the pairing can be removed and quickly replaced. Electrical connections are made with a standard RJ-45 connection for each pair, eliminating the need for tedious connections during pairing replacement. The base section and intermediate spacers between each T section are composed of Schedule 40 PVC pipe for rigidity and are not glued. Friction fits are strong enough for most joints and allow the sensor to be decomposed for storage and shipment in a single Pelican 1660 case [14].

3 Microcontroller coupled with vertically oriented detector elements

3.1 Microcontroller Hardware

The optical trip-wire profiling sensor has been shown to be an effective means of gathering silhouette profiles of subjects as they pass through its beams. Post-processing techniques have shown that these profiles can subsequently be classified into broad-scale classifications. Progression of the sensor development calls for the incorporation of data acquisition resources and profile classification algorithms into a single, field-ready package that does not require the use of a personal computer. Application scenarios require that the system be left alone with no user interface after initial setup. Thus, the requirement of a keyboard, mouse, monitor, operating system, and other requirements of a typical computer are no longer needed. A simple system capable of acquiring data, processing and classifying it, and reporting on its classification can be implemented using a microcontroller.

The microcontroller chosen for this particular implementation is a Rabbit® 4000 microprocessor. The microprocessor operates at 60 MHz and can support up to 16 MB of memory [15]. The microcontroller alone, though, requires an interface and is mounted to a BL4S200 single-board-computer (SBC) manufactured by Digi International ®. This particular system was chosen for its small size and capabilities in handling data acquisition, mathematical operations necessary for classification routines, memory, SD card onboard data storage, and network I/O capability. The SBC's native language of Dynamic C facilitates transition of the classification routines originally developed in Matlab to the embedded environment language. Though very similar to the traditional C language, Dynamic C is specially designed for programming embedded systems, and

features quick compile and interactive debugging [16]. The system is also somewhat scalable through the use of its RabbitNet interface, allowing for expansion to other boards of similar manufacture. The RabbitNet RN1600 expansion kit from Digi was similarly acquired to provide a low-level interface to the system during initial setup.



Figure 3-1. Digi® BL4S200 Single Board Computer [17].

Data lines from each of the 16 detectors in the near-IR profiling sensor are hard-wired directly into two 8-bit DIO ports on the microcontroller and are represented as 16-bit integer values. By representing an unbroken beam as a digital ‘1’ and a broken beam as a ‘0’, values from each port are combined to form an integer value using the equation:

$$P_x = \sum_{i=1}^8 s_i * 2^{i-1} \quad (3.1)$$

In equation (3.1), P_0 is the sensing element value from port 0 (representing the lower 8 optical trip line elements), P_1 is the value from port 1 (representing the upper 8 elements), s_i is the state for a particular detector, that is, an optical trip line element at position i within the upper or lower bank of elements. Values P_0 and P_1 are combined

using bitwise operations. The binary value of P_1 is upward shifted by 8 bits, effectively multiplying its value by 2^8 or 256. P_0 is then added to the shifted value of P_1 , forming a single 16-bit unsigned integer used to represent a particular time sample. This is a necessary step in minimizing the data requirements for implementation on system with limited memory resources. A single unbroken beam (optical trip wire) is represented as a 1 by the trans-receiver element; a broken beam is a 0. Thus, a single time sample with no unbroken beams will be represented by the value 65535, and a single sample with all 16 beams broken will be represented by a value of 0.

Each detector is wired independently from a common power source within the controller box. Although all detectors are currently powered from the same source, capabilities exist on the microcontroller to power each detector completely independently from the others, allowing the ability to terminate any particular detector from the microcontroller. This feature could be exploited in order to lower the system power requirements as well as to remove communication from a single errant detector. For example, if a problematic detector beam loses alignment with its mated reflector or is blocked by uncleared shrubbery, its power may be turned off at the microcontroller. Provided that adequate bi-directional communications exist between the microcontroller and a monitoring station, an operator could terminate power from his station to the single detector and restore the system to an operable state without having to make a call to the field. However, a minimal number of detectors should be powered down to avoid loss of data from passing objects of interest. Ground return lines are tied together. Russomanno and Chari have determined that profiles of interest can be correctly classified with a 98% success rate with as few as four detectors [2]. However, these four detectors are located

at specific heights. Arbitrary combinations of four detectors are not likely to recreate such results.

Data lines from each detector are also connected to a legacy 37-pin connection from the first profile sensor prototype in parallel, allowing for simultaneous data collection between the microcontroller and the prototype data collection system. Simultaneous collection is necessary to determine the microcontroller sampling rate. Laboratory experimentation with the microcontroller revealed that its data acquisition rate is significantly less than the PC-based USB data acquisition system, lowering the overall time resolution. Though normally an undesirable effect, the lowered resolution is actually advantageous in the case of a low-resolution microcontroller; lowered time resolution requires less memory resources required to store the data of a silhouette. The microcontroller's data acquisition rate is buffered by code written during development that indicates the sensor's status as it acquires a silhouette. This buffered rate was found by simultaneous collection of silhouettes by both the PC-based USB system and the microcontroller. Profiles generated by the two systems should be identical except for the number of samples in each; the profile acquired by the microcontroller's lower sampling rate will appear to be a compressed version of the PC acquired profile. A ratio is taken between the number of samples required to represent the profiles of the two systems. The CX-RVM5-PN detector element has a 1 millisecond response time [4], limiting the PC data acquisition system to a sampling rate of 1 kHz. With this known 1 kHz PC sampling rate in mind, the microcontroller's data acquisition rate is then simply a multiplication of the sample size ratios. This technique found that the microcontroller acquires data from all 16 of its detector input lines at approximately 21.3 Hz, far less than the 1 kHz system

utilized to acquire the library data with the original sparse sensor array prototype. Even so, this lowered sampling is sufficient to provide enough details for low-velocity subjects passing through the sensor's field of view. This lowered rate may also help eliminate noise introduced by inconsequential artifacts of the subject, such as a vehicle antenna, spare tire rack, horse lead, etc., while also reducing the amount of memory resources required. It should be noted that the microcontroller acquisition rate can be increased by eliminating the buffering caused by updating the sensor's status. Unbuffered, the microcontroller's data port collection rate quickly exceeds the memory resources of the system. Array overflows occur within 2 seconds of the start of data acquisition at this unbuffered acquisition rate. Lengthy and slow travelling objects passing through the detector quickly exceed this short time limitation and cause the microcontroller to fail.

The SBC hardware was packaged in an enclosure to provide protection from the elements during field data collection events. This enclosure contains connections for the simultaneous data collection by both the microcontroller hardware as well as the PC-based system. The interior of the enclosure was lined with aluminum tape in an attempt to eliminate problems associated with electro-static discharge. It should be noted that the addition of this aluminum tape may cause cooling problems when fielded in hot summer conditions if no cooling is added; additional testing is warranted to determine if additional cooling will be required.

The SBC and profile sensor array of 16 detectors draw approximately 800 mA (1000 mA peak) of current at 12 VDC. Power is supplied to the system via a N-sized 12 V DC connection on the SBC enclosure. Laboratory and field collections have utilized a

common 110 VAC to 12 VDC transformer, although other possible sources such as batteries or solar panels could also be utilized.

3.2 Feature Extraction and Classification

Several classification algorithms have been analyzed in prior works for the profiling sensor. However, as previously discussed, these algorithms were executed off-line and were not limited by the resources of the host computer. However, for this implementation, the design of the classification algorithm is constrained by the requirement to implement it on a low-resource platform such as the Rabbit BL4S200 embedded controller. The single feature of height/width ratio works well for discriminating humans versus other objects, but fails to make a proper discrimination between animals and vehicles. Figure 4-1 illustrates the height and width features of each of the samples within the time based sample library [7]. These two parameters form the single height/width ratio.

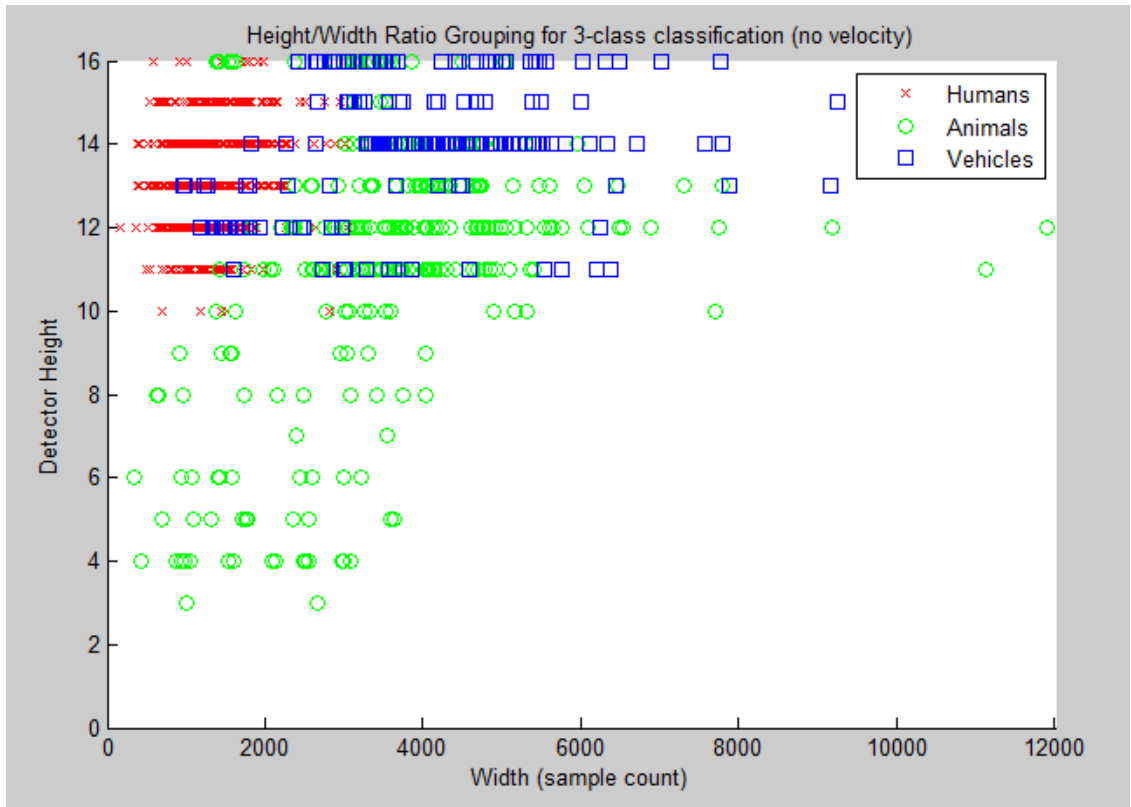


Figure 3-2. Height/Width Ratio feature for samples in the Profile Sensor 'Timing' library.

Coupled with velocity calculations to create an accurate width, the height/width ratio has provided a 99% classification rate with a pyroelectric sensor [8]. However, without incorporation of velocity, the width of an acquired sample is the number of the time samples required for the subject to pass through the sensor. Since this time t is a function of both the physical width d of the object as well as its velocity v as expressed by equation 3.2,

$$t = d/v \tag{3.2}$$

the average velocity must be known to make a reasonable estimation of the profile's width. However, with the vertical array alone, velocity calculation is impossible.

The height-to-width feature has been demonstrated to distinguish between humans and non-human objects with a 90% classification rate [2]. However, the feature fails to properly distinguish between vehicles and animals. A more robust feature extraction method, referred to as the six-feature technique, was implemented to make the distinction between animals and vehicles.

The six-feature extraction technique operates by centering an acquired profile into a ‘box’ of fixed width. The fixed width of the box is defined by the longest sample length within the signature library. The height of the box is tailored to the maximum height of the subject. The box is then partitioned into six sections by horizontal halving and by making two vertical partitions at 1/3 and 2/3 the length of the box as shown in Figure 4-2. The number of ‘off’ instances in each of the six boxes is counted and normalized by the highest value count among the six sections. This process generates six features for each profile.

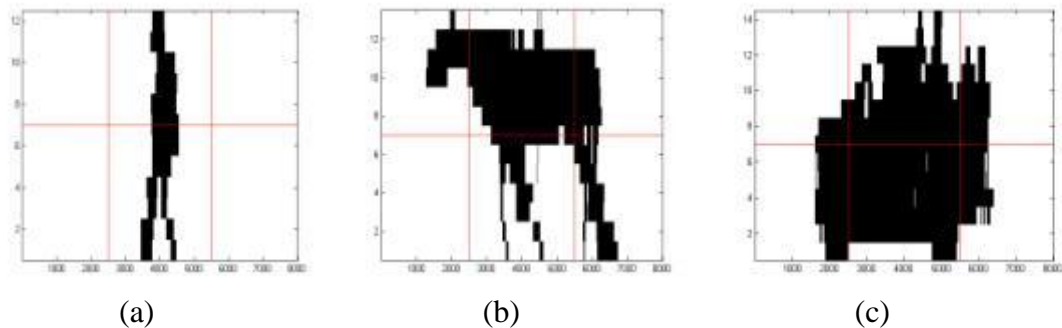


Figure 3-3. Human, animal and vehicle profiles and derived feature spaces.

Note that the width of the six-feature box is different between the PC-acquired higher sampled library data and slower microcontroller rate of 21 Hz. This collection rate difference can be accounted for by defining a smaller ‘box’ for the slower-rate SBC

collected data. The size ratios of the two box widths are the same as the acquisition rates. For example, a box of 10,000 samples recorded at 1 kHz represents 10 seconds of data. To record the same 10 seconds at 21 Hz on the embedded controller, a box size of only 210 is required. This size difference is accounted for during the pixel count normalization process of the six features. Although down-sampling the library data will yield profiles of approximately the same length as those collected directly by the slower SBC, this is an unnecessary computational step. Values will be normalized by the highest pixel count no matter the length of the six-feature box.

The statistical means and standard deviations of the six features for each class are then computed from the training data set. When a test object moves through the profiling sensor, the Naïve Bayesian distance between the test object and each of the three classes is computed in the feature space. The Naïve Bayesian distance between a six-feature test sample and the K^{th} class is represented by equation (3.3) [18].

$$NB_K = \sqrt{\sum_{i=1}^6 \frac{(\mu_{Ki} - t_i)^2}{\sigma_{Ki}^2}} \quad (3.3)$$

NB_K is the Naïve Bayesian distance to each class K , μ_{Ki} and σ_{ki} are the means and standard deviations for each of the six features within their class K , respectively, and t_i is the test sample. The test sample is assigned to the class which has the smallest Naïve Bayesian distance with respect to the test sample.

It is also noted that this relatively simple feature extraction technique is specifically suited to the type of terrain described in the introduction section and shown in Figure 1-1. The narrow and rugged pathways in such terrains only allow for the objects

to move at slow speeds in confined spaces. Since the objects move between the trans-receiver and reflector platforms, which are placed not more than 15 feet apart, effects of profile height variations due to range (distance from Trans-Receiver to object) are negligible.

3.3 Classification Results

The microcontroller's classification algorithms were tested with the vertically oriented sparse sensor array against a sample set of animals and humans. During this data collection event it was realized that the microcontroller's sampling rate had been miscalculated during laboratory experimentation. An initial, incorrect sampling rate of 60 Hz was used in defining the maximum profile length and division lines of the six-feature classification routine. Later experiments revealed that the sampling rate to be 21 Hz. Profiles acquired with the slow sampling rate are much smaller, allowing them to easily fit within the middle sections of the six-feature classification routine. As a result, most all profiles acquired were improperly classified as human when processed in real-time.

4 Custom Sensor Array

4.1 Introduction

The active IR sensing elements require that a subject pass through the beams to be detected. This requirement limits the OTW to deployments in which objects of interest must pass through a very constricted area. Individual detectors must be arranged various heights, the highest of which must match the object to be detected. Thus, the overall size of the sensor is quite large, making the system conspicuous and thwarting its concealment. Requiring a subject to pass through the sensor to obtain its profile presents a limitation that is easy to counter. If a person recognizes the sensor that is being used to monitor him/her, he/she merely needs to walk around the sensor to avoid detection. Good concealment is therefore essential. Part of the concealment may be done by breaking the vertical array apart, and distributing the detectors along an anticipated path of objects of interest; thereby, reducing the obtrusive size of the single array pole and its reflector [2, 7].

Detectors of the profiling sensor are traditionally placed in a sparse vertical column configuration. Since no horizontal spacing exists between the sensing elements, timing between the detectors is synchronized. Acquiring a profile from the detector elements that are not in the same plane requires that the precise locations of the detector elements be known to synchronize timing between them. Measuring these distances by conventional techniques is particularly time-consuming, especially if the sensor elements are to be placed at significant horizontal distances from each other. Long setup/measurement time makes the custom configurations impractical for deployment in

the field, especially when deployment time is minimal; a quick method of measuring the distances between the profiles is therefore required.

4.2 Hardware Configuration

Experimentation for the custom sensor array was realized by placing the detector locations at specified heights and random horizontal locations. This setup was to be constructed using a 4'x8' pegboard as illustrated in Figure 4-1. The 1" grid pattern of the pegboard gives a readily available indexable setup pattern for sensor placement. Detectors are placed at random horizontal locations on rows that are vertically separated 5" from each other, consistent with the 5" separation between detector elements of the column array profile sensor. While good for experimentation within a laboratory environment, the 4'x8' pegboard implementation is not conducive for field implementations. Not only is transportation difficult, but animals traditionally studied during field collections (cows especially) would be leery of passing beside a 4'x8' wall they are not accustomed to.

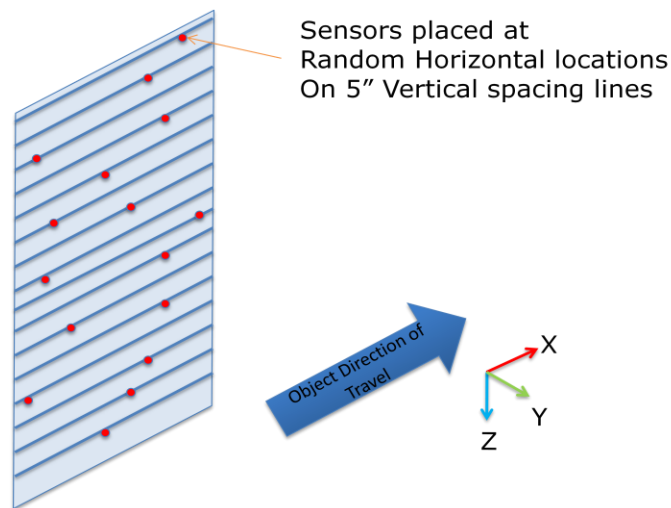


Figure 4-1. Prototype custom array utilizing a 4'x8' pegboard.

A more suitable implementation was realized by modifying the single-column PVC array. The modularity of the PVC single-column array lends itself easily to the implementation of a broken array due to its construction of separable sensor elements. The eight sensor pairs are removed from the column array and placed on a section of PVC pipe to set them at their original vertical heights as illustrated in Figure 4-2. Horizontal spacing is random. Note that this particular configuration is a prototype with horizontal distances of approximately 15-35 cm between detectors. Actual field implementation would place individual detectors where maximum concealment can be achieved, possibly with significantly greater horizontal separation.



Figure 4-2. Example of custom array implementation using PVC pipe.

Each near-IR break-beam detector still requires its own mated reflector. Reflectors are placed on PVC pipe spaced at the same random horizontal distances as their corresponding detectors. Thus, beams of the 16 detectors are parallel with one another across a three-dimensional space. For this particular prototype, reflector poles

are made the same length as the spacer poles used to hold up the detector pairings, allowing interchangeability.

Profile acquisition has traditionally been done by recording data only when a subject is within the beams of the near-IR detectors. Logging begins at the first beam break and terminates when all 16 beams are continuous again, where it is assumed that the passing object is completely through the sensor array. While this approach works for the traditional array where all 16 detectors lie on the same column, it fails when the elements are removed from the array. Since a horizontal distance between the detectors is introduced by removing the detectors from the vertical column, it is possible for a subject passing through the detectors to fit 'between' two elements, terminating the logging. This problem is solved by logging a determined number of additional samples after the beams of all 16 detectors are continuous. If a detector beam is broken within this period, logging continues. Otherwise, logging terminates. Since no profile data lies within the additional sampling period (because all the detector beams are continuous) the additional sampling period is removed from the profile.

4.3 Simple classification technique using custom array

A simple classifier can be implemented using the custom array by summing the 'on' and 'off' states of the detector along each row as an object passes through the sensor. Energy along an individual row is considered where a detector beam is broken by an object. Placement of energy along a row is independent of the placement of energies of other rows. The summation data along each row is normalized by the greatest summation value of the 16 detectors. These normalized values express the relative amount of energy along the rows, where energy is considered to be made by a broken beam. These

normalized values for each of the 16 rows are treated as independent features and compared to training data sets of human, vehicle, or animal. The object is subsequently classified using Naïve Bayesian classifier. Throughout the remainder of this paper, this technique will be referred to as the “Row Energy” technique.

An interesting feature of the Row Energy technique is that the energy normalization along each row eliminates the necessity to consider the overall length of the profile. The normalized profile of a slow moving object should be identical to the normalized profile as if the object were moving quickly. This allows the row energy technique to be velocity independent. Note that the normalization causes the width of all profiles to be equal; the particular rows that the profile’s normalized energies lie upon are the features under consideration. Also note that this normalization also allows a comparison to be made between the library of acquired samples to those attained by the lower-data sampling rate of the microcontroller. Normalized values for each of the 16 rows are displayed graphically in Figure 4-3 for the broad classifications of humans, animals, and humans. Standard deviations for the normalizations are displayed by error bars along each value.

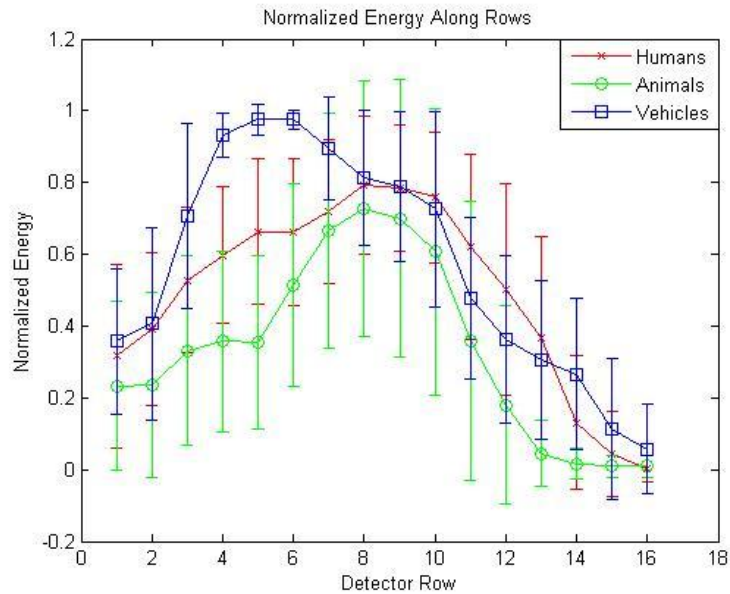


Figure 4-3. Normalized energy along rows for samples of the profiling sensor library.

A leave-one-out classification study against the profile sensor’s acquired sample library [7] revealed that the row energy technique can obtain a 92% classification rate with a Mahalanobis distance classifier. Its ease of implementation makes it particularly appealing for a low-resource microcontroller. However, the technique fails to calculate the object’s velocity or its direction of travel, both of which may provide valuable information on a passing subject. Higher-yielding classification algorithms have been developed for profiles generated by the sensor with vertically oriented detectors [2, 6, 7]. However, to utilize these algorithms, the profile must be reconstructed to appear as if it were generated by the sensor with vertically oriented detectors.

4.4 Custom array profile reconstruction technique

Detector timing for the vertical column configuration is synchronized by the placement of the detectors along the same column, as illustrated by Figure 4-4. Most subjects passing through the sensor’s field of view do not have a flat leading edge; therefore, their profiles

will illustrate a non-straight edge. Passing a straight, vertical object through the array will cause all 16 detectors to trip simultaneously, as illustrated by Figure 4-5.

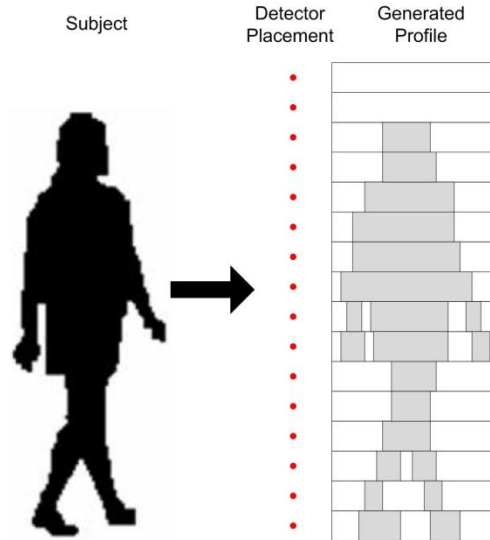


Figure 4-4. Subject passing through a vertical column sensor.

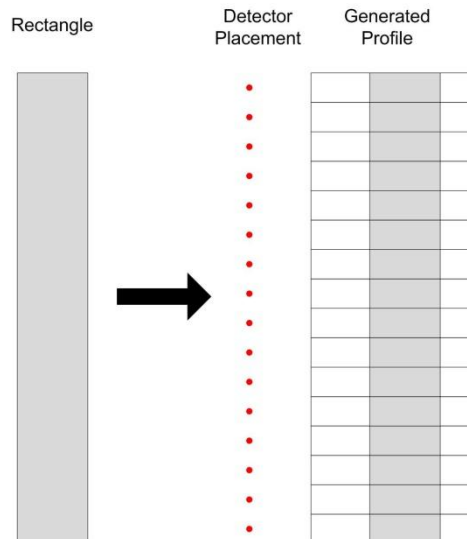


Figure 4-5. Straight vertical object passing through vertical array.

Removing these detectors from the same column destroys the synchronization between them. Passing the straight vertical object through an array in which the detectors are not vertically co-located will create a profile in which the detector elements

‘on’ and ‘off’ times are modified by the placement of the detectors. Figure 4-6 illustrates the passing of a vertically straight object through a modified array. Passing this rectangular ‘calibration rectangle’ through the sensor is done as part of a calibration routine to define the distances between each detector pair. Since it is known that the leading edge of this rectangular object is vertically straight, the physical location of the detectors can be determined by counting the number of time samples between a detector pair’s first transition state. For example, if there are X samples between detectors on row i and row j , then it can be assumed that the physical distance between detectors on rows i and j for subsequent samples should be offset by the same X number of time samples. The number of samples between rows is found by searching for the first transition state of each row from ‘on’ to ‘off’, corresponding to the leading edge of the passing rectangle. Similarly, the trailing edge may also be used if the calibrating vertical object has a vertical rear edge. Note that the vertical calibration rectangle must be passed through the array at a relatively constant speed to trip the elements at their correct physical locations.

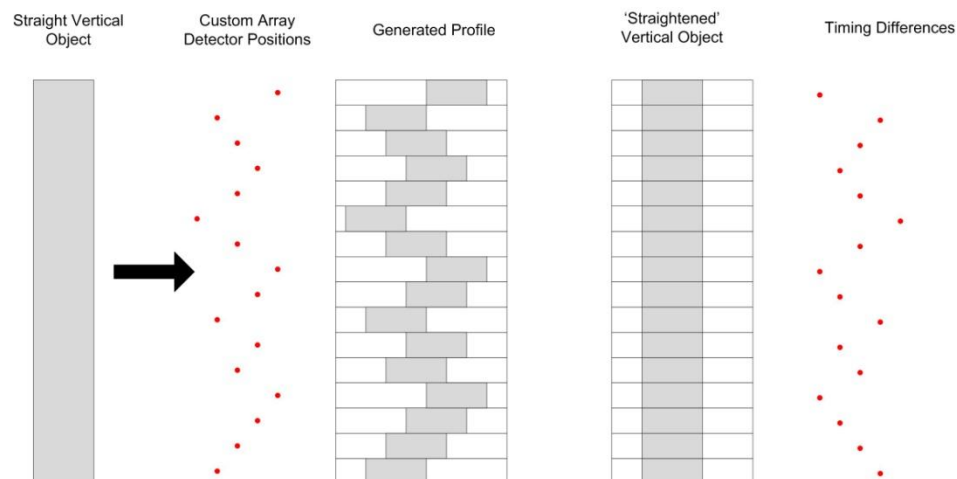


Figure 4-6. Straight vertical object passing through custom array.

These time differences (corresponding to the positions of the individual detectors) are then subtracted from the raw timing profile of each subsequent subject passing through the array to rebuild the proper profile. Figure 4-7 illustrates a subject passing through the prototype custom array and its subsequent reconstruction.

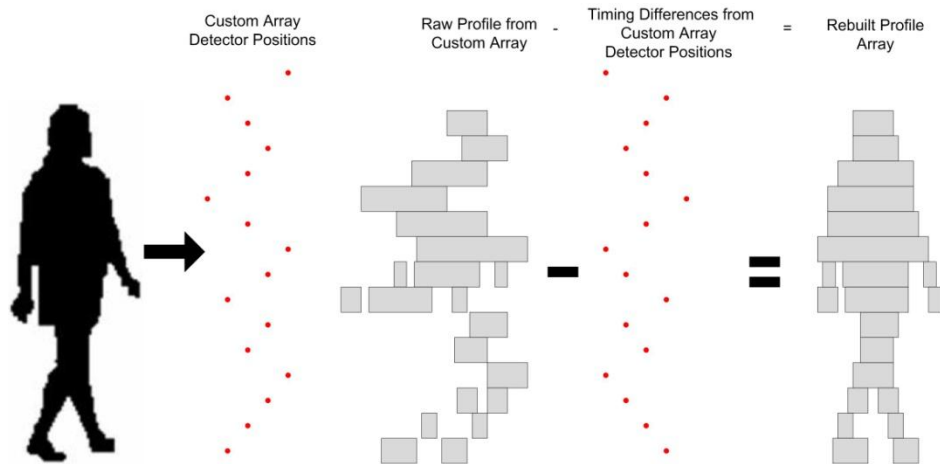


Figure 4-7. Subject profile reconstruction after passing through broken array.

This technique assumes that, once in the field, all objects passing through the detectors of the sensor array will pass through the entire horizontal distance of the array and travel at a relatively constant speed.

It should be noted that the heights of the detectors must be known before this calibration routine is conducted. Detectors are arranged in ascending order; detector 1 is at the bottom and detector 16 is at the top. For purposes of this study, these detectors have hard-wired connections to the microcontroller. However, it is possible to dynamically assign the locations of detectors at various heights. A triangle, oriented with the base parallel to the ground, passed through the array will generate a profile where the lowest detector is engaged the longest. The highest detector will be engaged the shortest period of time. Detectors are then dynamically assigned their positions

based upon their ‘on’ times. This tool may be a valuable feature for quick deployment where detectors are arbitrarily placed with no particular attention to the heights of the detectors.

4.4.1 Implementation of Profile Reconstruction

As an illustration for this section, a custom detector arrangement was made in which the detectors are in a roughly diagonal configuration as shown in Figure 4-8. Note that detector placements do not necessarily require such an arrangement, but may be arranged in any particular fashion.



Figure 4-8. Custom profile sensor array prototype realized on PVC pipe.

Also note that the element pairs do not necessarily have to lie on the same vertical plane: detectors in Figure 4-8 were left paired together, due to their construction. Figure 4-9 (a) illustrates a human carrying a 2x4 vertically through the sensor (moving from right to left through the sensor in Figure 4-8) as part of a calibration routine. The calibration routine determines the first transition state of each detector and finds the time

difference between the profile's first detector's trigger event and the first trigger event along each row. By subtracting these time differences along their respective rows, the profile is 'straightened' as shown in Figure 4-9 (b). Note that the same 2x4, shown in the oval, is shown again in Figure 4-9 (b) in a more recognizable format.

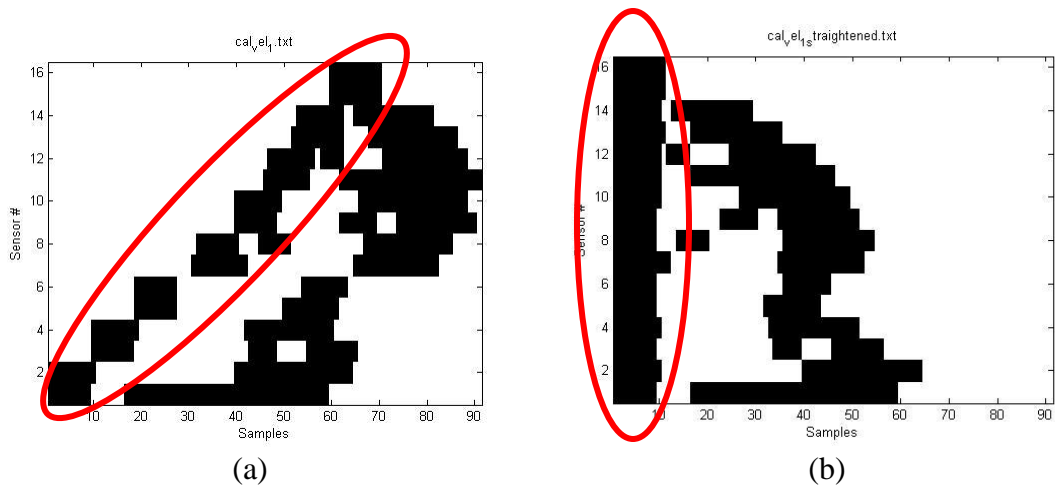


Figure 4-9. Person carrying a vertical object through the custom sensor array of Figure 4-8; (a) uncorrected and (b) corrected. Vertical beam is shown in oval.

Subsequent data acquisitions have this same row shifting routine performed on them. Figure 4-10 shows the corrected profile of a person passing through the array without the 2x4 used for detector timing calibration.

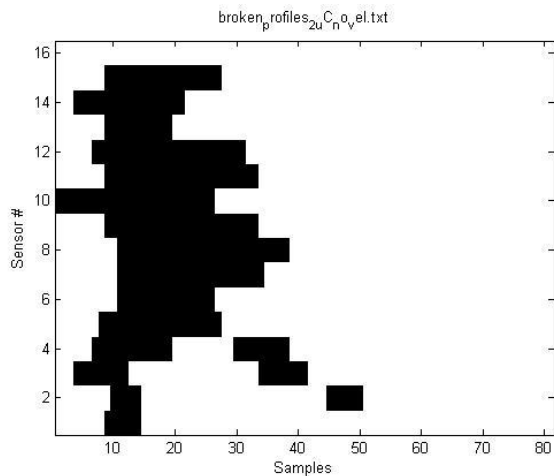


Figure 4-10. Corrected data sample collected through custom array.

This technique of shifting the start position of each of the rows appears to account for the random horizontal placement of detectors. Although the array may be placed in any customized configuration, the specific detector locations must remain fixed after the calibration routine is executed. If the array is modified, the calibration routine must be performed again to find the detectors' positions relative to one another.

Note that data collected from the vertical column array profile sensor consists of timing samples. A slowly travelling subject will generate a profile of greater width than if it were to pass through the sensor quickly. Thus, the number of time samples used to generate a profile alone is not an accurate measurement tool to indicate the physical width of the passing subject. Merely applying the time sample shifting to each of the rows is not an entirely accurate method of realigning a profile since subjects may pass through the array at speeds other than which the detector distances were calibrated. Since $t = d/v$, the amount of shift t applied to each row is a function of the element distances d and the velocity v of the travelling subject, not merely a count of the number of time samples between rows. Subjects with a speed v , which may be different than that of objects used in the calibration, will, therefore, have malformed rebuild profiles as illustrated in Figure 4-11 (a). Furthermore, correcting the profiles due to time only does not account for the direction of travel that a subject may take through the sensor. Travelling in the direction opposite from what the array was calibrated causes the row elements to be shifted in the opposite direction as shown in Figure 4-11 (b). Although this technique of profile rebuilding may work appropriately if we make the assumption

that subjects will always pass at the same speed, it fails when the assumption is violated. A more robust calculation technique is therefore required.

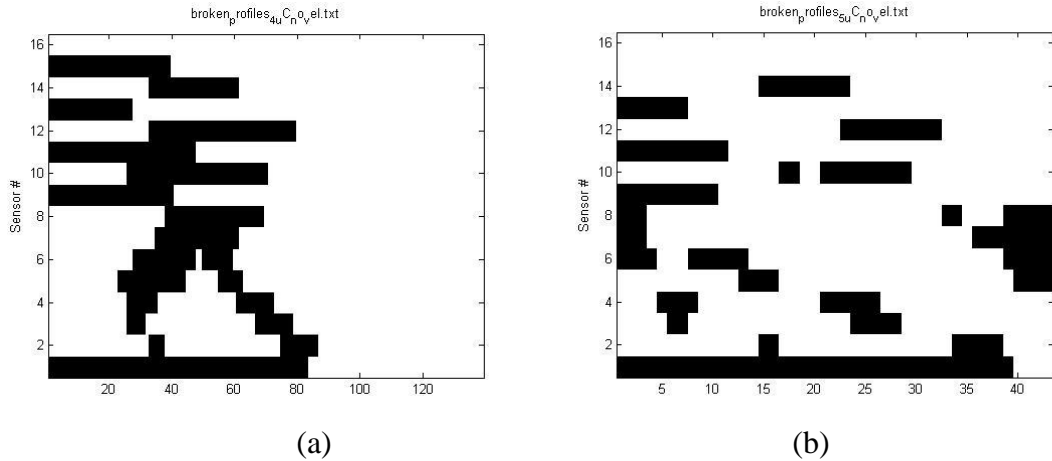


Figure 4-11. Incorrect 'Corrected' profiles.

4.4.2 Velocity calculation and incorporation

A more accurate method of correctly rebuilding the profiles requires that the physical distances between detectors and the actual velocity of the passing subject be determined. Using these parameters, a more accurate time shifting value may be calculated for each detector state. Physical distances between the elements are determined in an identical fashion as described earlier. However, the distance between any one pair of elements must be known. Ratios of the numbers of time samples recorded between element pairs are computed. Assuming that the calibration rectangle is passed through the sensor at a relatively constant speed, the physical distances are a product of the time ratios between the elements. These distances are calculated several times as part of the calibration routine and averaged along each row. Figure 4-12 represents the positions of the detectors for the custom array configuration of Figure 4-2 across multiple passings of the calibration rectangle at a variety of speeds.

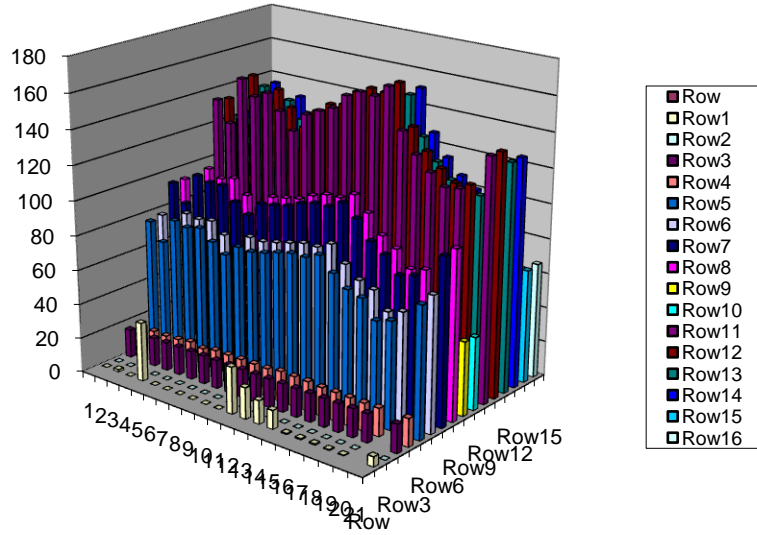


Figure 4-12. Calculated detector positions across multiple calibration passes.

The average position of each detector is calculated from multiple passings of the calibration rectangle through the array. This allows for slight discrepancies generated by the calibrating rectangle passing through the array at a non-constant speed.

A subject's velocity through the profile sensor is calculated by averaging the individual velocities v_{ij} generated between every possible element pair along the leading and trailing edges of the profile [10]. This mean velocity is given as the expression

$$mean\ velocity = \frac{2}{M(M-1)} \sum_{\substack{i=lowermost \\ detector\ trigger\ event}}^{M-1} \sum_{j=i+1}^M v_{i,j} \quad (4.1)$$

Where $i \neq j$, and M is the uppermost detector trigger event. Velocity between each detector pair is defined as $v_{ij} = d_{ij}/t_{ij}$ where d_{ij} is the physical distance between elements i and j , and t_{ij} is the number of samples between the elements i and j to the leading or trailing edge of the profile. Although seemingly computationally intense, the number of

element pair velocity calculations is limited to $M*(M-1) / 2$, where the maximum value of M is the number of detectors in the array. This leaves a maximum of only 120 possible calculations along each edge for a 16 detector array. Combinations where $t=0$ cause division by zero and are not considered. The implementation of the vertically paired sensor detectors causes horizontal pairings to have a time difference of zero, thereby eliminating those particular element pair velocity calculations from consideration and further reducing the computational requirements of the microcontroller.

A rectangular-shaped object passing through the sensor would generate constant element pair velocities. Upright, walking humans are mostly rectangular in shape, generating similar velocities along the horizontal element pairings for the front and rear edges of the profile. However, not all profiles share this rectangular pattern. The overhanging head of an animal, sloped vehicle windshields, swinging arms, etc., caused by premature detector triggering, can generate abnormal detector pair velocity calculations. These particular velocities are eliminated by setting a threshold of a maximum reasonable velocity.

Once an overall subject velocity is calculated, the amount of time required to shift each row t_{ij} is calculated in samples by $t_{ij} = d_{ij}/v$, where d_{ij} is the physical distance between the sensor elements i and j found from the calibration routine. Note that this velocity calculation is capable of generating both positive and negative values; with this data a direction of travel through the gates can be determined. Figure 4-13 illustrates the more properly reconstructed data of the same profiles found in Figure 4-11 by this more accurate technique.

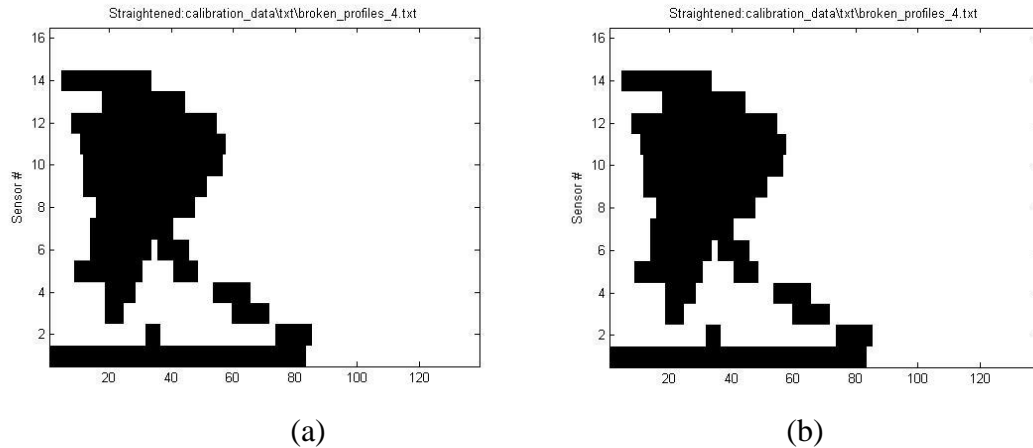


Figure 4-13. More accurately reconstructed profiles of Figure 4-11 generated by velocity incorporation.

Incorporation of velocity allows a more accurate profile width to be determined from data acquired from profiling sensors [10]. Classification algorithms, such as the height/width ratio and six-feature technique discussed in section 3, are based strongly on the width of the sample. Improving the determination of the width of an object will improve the overall classification of objects by profiling sensors. Since the velocity is calculated for rebuilding profiles with the custom array, it can also be utilized to calculate a more accurate ‘width’ for the profile. Each sample of time within the reconstructed profile is duplicated by the velocity value, creating a secondary profile that more accurately represents the physical width of the object. This secondary array is referred to as a “distance” profile.

Note, however, that the value of the velocity is a non-integer number. Slow moving objects under consideration typically generate low valued velocities. Simply rounding the velocity to the nearest integer is not an accurate means of determining its width. For example, consider velocity values of 1.4 and 1.6 for a particular profile. If rounded down to 1, the resultant distance profile would be only half the size as if its

velocity were rounded up to 2. This problem is solved by non-integer sampling [19]. Each sample of the time profile is upsampled by a factor of five times the velocity value into a secondary array. The secondary array is then downsampled by a factor of five, generating the distance profile. Upsampling and downsampling by a factor of five allows the velocity to be multiplied in increments of 0.2. A higher upsampling/downsampling rate of 10 was attempted. However, it was found that the higher quickly rate caused array overflow problems on the low resource microcontroller.

It should be noted that the resulting widths of the ‘distance’ profile are expressed in the same physical distance units that the individual detectors are expressed. Physical distances between the detectors in this research are expressed in cm. Generated distance profiles are a product of their raw profiles (expressed in a count of time samples) and their velocity (expressed as cm travelled per count of time samples). Therefore, if a torso of a profile is 20 samples wide, it represents a torso of a passing object approximately 20 cm in width.

4.5 Results

Algorithms discussed previously in this paper have been implemented utilizing the Rabbit BL4S200 microcontroller discussed in section 3. Data collection, velocity calculation, and profile reconstruction tasks are completed in real-time for data collected with the custom configuration of the active near-IR profiling sensor array. Real-time classification is also performed.

Figure 4-14 illustrates a histogram of calculated velocities for a variety of objects captured during a data collection event. Note that both positive and negative velocities are shown. Negative velocities represent travelling through the array in an opposite

direction. Also note that most objects studied travel at walking speeds; outliers are made by faster moving vehicles. Maximum vehicle speeds were kept under 5 mph due to the hardware configuration along a narrow roadway during the data collection event.

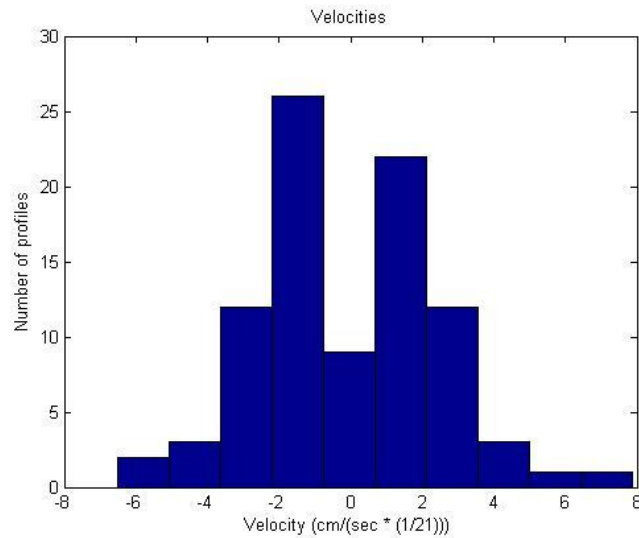


Figure 4-14. Histogram of object velocities from a data collection event.

As stated previously, the computed velocities are not expressed directly in the form of distance per unit of time, but rather in the form of distance per samples collected. However, since it is known that 21 samples are collected per second, the computed velocity values can be quickly transformed into a more meaningful unit such as meters/second or miles/hour. Thus, for a subject's calculated velocity of 2 cm/(samples * sec/ 21), this velocity is more conventionally expressed as 0.42 m/s, or 0.93 mph. This follows the data collected during the field study; most humans and animals walked through the profile sensor array at a slow pace of approximately 1 mph.

Figure 4-15 (a) illustrates the raw data of a profile as it is acquired with the custom array of Figure 4-2. The real-time reconstructed profile is shown in Figure 4-15 (b).

This particular object creating the profile is a human with a large pack. Note how the

reconstructed image appears to take on the appearance of similarly looking human profiles acquired by the linear array. This rebuilt profile is now ready to be analyzed by classification routines such as the height-to-width ratio or six-feature extraction technique.



Figure 4-15. Raw data acquired from custom array (a) and Reconstructed Profile (b).

The reconstructed distance profile is representative of the object's width. For example, if the profile has X number of samples along a particular detector row in the 'off' state, represented by a '0', then the object is likely to be X cm wide at the height of that particular detector. Although various features can be extracted from the profile such as the object's height-to-width- ratio or the normalized six-feature values, to date, there are few samples within the library of profiles that have the physical width derived based on the subject's velocity. Comparison of the physical width of a recently acquired sample to the timing width of samples within the "timing" library acquired by the profile sensors with the vertical array is likely to lead to fallacious results.

The simple Row Energy technique is currently utilized as a classification tool on the microcontroller until future data collection events can be performed with the custom array. Detector state events are summed and normalized by the highest number of detector events of the 16 row features, making this technique particularly appealing: the

“timing” library may be compared against the raw timing features of a subject test case. Field and laboratory experiments have yielded an 85% classification rate for humans, vehicles and animals when applied to a Naïve Bayesian classifier trained against samples of the ‘timing’ library. Real-time classification results for the Row Energy classification technique are displayed in the confusion matrix of Table 4-1.

Table 4-1. Real-time Row Energy Classification Confusion Matrix

| | | True Classification | | |
|--|---------|---------------------|-------------|--------------|
| | | Human (30) | Animal (32) | Vehicle (29) |
| Real-time Row Energy Feature Classification | Human | 83% | 16% | 0% |
| | Animal | 0% | 100% | 0% |
| | Vehicle | 3% | 22% | 72% |

It should be noted that classifications are made upon the complete profiles. Profiles of most animals studied were led by a human. The distance between the human leader and the following animal was not sufficient to have only one subject in the customized profiling sensor. Thus, the captured profile contains both the human and the following animal. For purposes of this study, these acquisitions are considered to be animals until the human profiles can be removed.

Data collection events have yielded custom array profile data on 30 humans, 32 animals, and 29 vehicles. Example custom array profile acquisitions of humans, animals, and vehicles are displayed in Tables 4-2 through 4-4, respectively.

Table 4-2. Example Human profiles, Real-Time Reconstruction



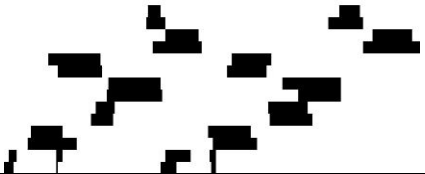
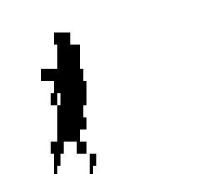


| | Human waking North → South | Human waking South → North | Two Humans |
|-----------------------|---|---|---|
| Raw Data |  |  |  |
| Reconstructed Profile |  |  |  |

Table 4-3. Example Animal Profiles, Real-Time Reconstruction







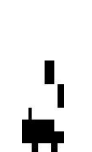





| | Large horse and human leader | Medium horse and human leader | Small Dog | Medium horse and human leader |
|-----------------------|---|--|---|---|
| Raw Data |  |  |  |  |
| Reconstructed Profile |  |  |  |  |

Table 4-4. Example Vehicle Profiles, Real-Time Reconstruction

| | Midsize truck, windows down | SUV, front windows down |
|-----------------------|---|--|
| Raw Data |  |  |
| Reconstructed Profile |  |  |

Although completely implemented, note that the profile rebuilding process is not entirely correct. Additional ‘noise’ occurs on many of the rebuilt profiles. This noise is due to a flaw in the real-time row re-alignment phase of the profile rebuilding process. This noise does not appear in the raw data directly acquired from the custom array, but introduces itself as part of the realignment process. Figure 4-16 illustrates the introduction of this noise during the realignment process for the profile of a small dog.

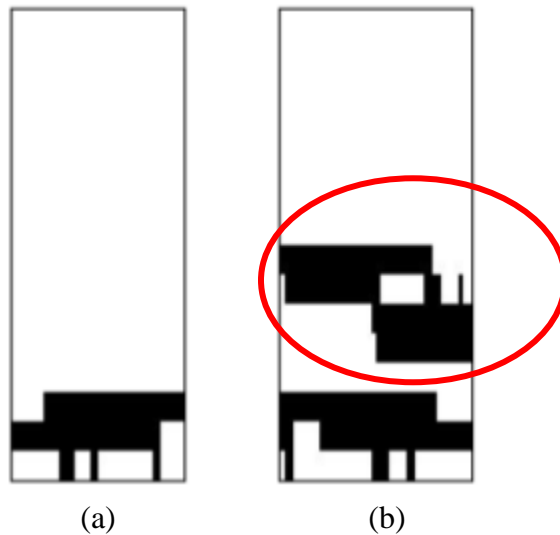


Figure 4-16. Raw (a) and rebuilt (b) profile of a dog illustrating noise introduction (circled in red).

Not all of the profiles processed in real-time suffer from this noise problem. However, those that do are likely to be misclassified by the height-to-width and six-feature algorithms. Future work involves repairing the profile rebuilding algorithm so that these classification algorithms may be implemented. Note, however, that even with the added noise, the microcontroller is still capable of producing a real-time 85% classification rate using the row-energy technique.

Data from the collection events discussed earlier was processed offline with Matlab through the same algorithms. It was discovered that the noise problem is due to a problem with algorithm's implementation in C; noise is absent from the profiles. Figure 4-17 demonstrates the correctly rebuilt profile of Figure 4-16 without noise.

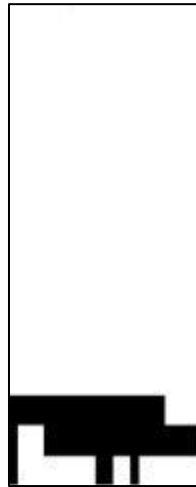







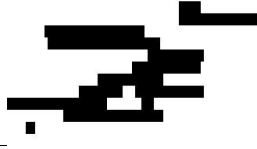






Figure 4-17. Rebuilt profile of Figure 4-16 without noise.

While most profiles are rebuilt properly, improper velocity calculations can cause malformed rebuild profiles. Improper velocities cause data along the rows to be shifted by an improper amount, malforming the rebuilt profiles, and giving them lower aesthetic qualities than the raw profiles used to generate them. Table 4-4 illustrates raw profiles and their reconstructed profiles generated for a variety of objects.

Table 4-5. Post-Processed profiles demonstrating poor rebuilding

| Description | Raw Profile | Reconstructed Profile |
|---------------------|---|---|
| Human leading horse |  |  |
| Human leading horse |  |  |
| Human |  |  |
| Human |  |  |
| SUV |  |  |
| Small Truck |  |  |

Height-to-width ratios were also calculated for all profiles generated by the custom detector array for an offline classification study. Real-time classification results were not possible, because prior to the data collection event, there were no values to compare the profile's height to width feature to. A take-one-out classification study allowed a comparison to be conducted. This take-one-out study revealed that the height to width feature had a classification rate of only 54%. This low classification rate is

likely caused by poor velocity calculations, resulting in poorly generated profile widths. The classification confusion matrix for the height/width ratio feature is displayed in Table 4-5.

Table 4-6. Height/Width Feature Confusion Matrix

| | | True Classification | | |
|--|---------|---------------------|--------|---------|
| | | Human | Animal | Vehicle |
| Height/Width Feature Classification | Human | 83% | 13% | 3% |
| | Animal | 13% | 16% | 79% |
| | Vehicle | 20% | 9% | 69% |

A similar take-one-out study was conducted using the six-feature box technique discussed in section 3. The percentage of correct classification is subject to the size of the ‘box’ that the profile is placed into as well as the dividing lines that divide the profile into six individual areas. Earlier studies had placed the dividing lines at arbitrary locations of 1/3 and 2/3 the width of the total box length, where the box length is the length of the longest sample within the library. In order to optimize classification results, a more specific divider location is required. Locating these specific locations is performed by conducting multiple take-one-out studies using the library of rebuilt profiles while varying the box width and divider location lines. Divider line locations are defined by their distance from the center of the box housing the centered profile. A study was conducted in which the divider lines were varied from 1 to half the width of the box, where the width of the box was varied from 100 to the largest sample size of the library. Results from the multiple take-one-out studies are shown graphically in Figure 4-18:

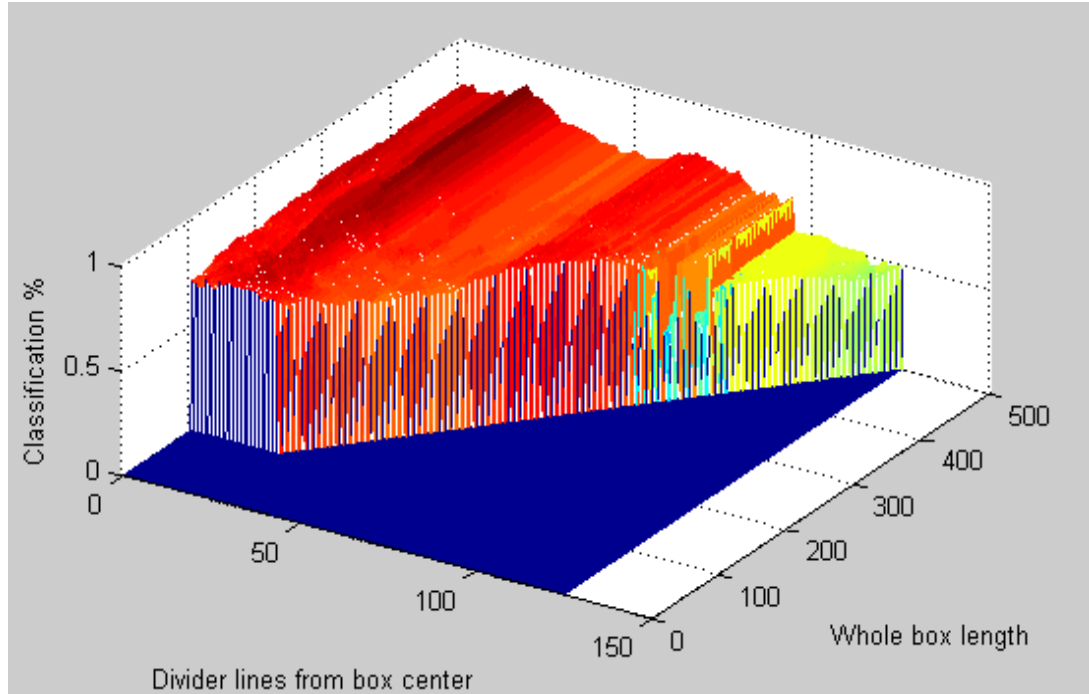


Figure 4-18. Six-Feature classification rate % dependent upon box width and divider placement from center.

It was found that classification rates do not vary greatly with the size of the box. For consistency, the box width was set to slightly larger than the width of the largest sample of the reconstructed profiles. Divider locations from center, though, have a large effect on the classification. Figure 4-19 illustrates the peak six-feature classification rate of 86% for dividers placed at +/- 19 samples from the center of the profile.

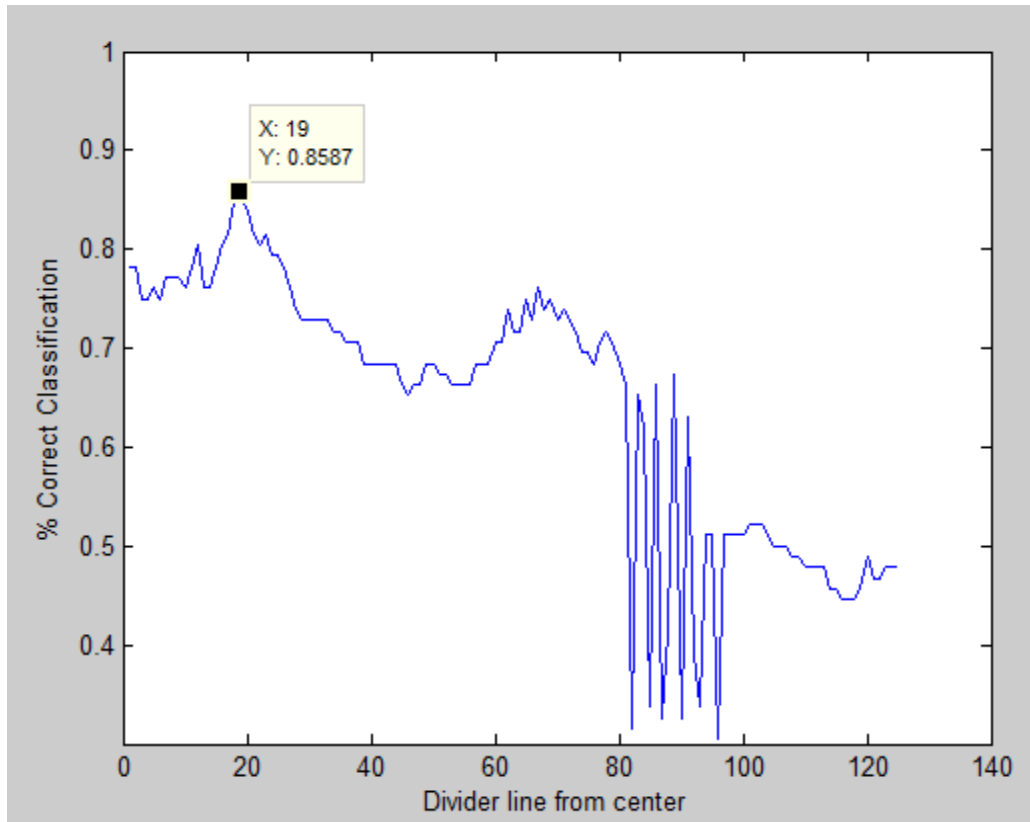


Figure 4-19. Six-Feature classification rates for box width 500, demonstrating peak classification rate.

Note that for this application, each sample represents 1 cm. Placement of the six-feature divider lines at ± 19 samples from the profile center makes sense; these dividers are spaced 38 cm apart from one another. Most humans, walking upright, should be no more than 38 cm wide. Large animals and vehicles should quickly exceed this limit. Even so, small animals that were smaller than this 38 cm threshold were considered as part of this study. Their improper classification as humans lowers the overall classification rate of the six-feature routine. Individual classifications for the Six-Feature classification technique are displayed in the confusion matrix of Table 4-6, where the vertical divider lines are spaced at ± 19 cm from the profile center.

Table 4-7. Six-Feature Classification Confusion Matrix

| | | True Classification | | |
|----------------------------|---------|---------------------|--------|---------|
| | | Human | Animal | Vehicle |
| Six-Feature Classification | Human | 87% | 13% | 0% |
| | Animal | 7% | 87% | 7% |
| | Vehicle | 3% | 9% | 86% |

Table 4-8 summarizes the post-processed classification rates for realigned profiles. The Mahalanobis distance classification algorithm is not implemented on the microcontroller; future work involves implementing this classification algorithm.

Table 4-8. Classification rates for profile features

| | Feature | Offline, Naïve Bayes | Offline, Mahalanobis |
|--------------------------|--------------------------|----------------------|----------------------|
| ‘Time’based library | Height/Width Ratio | 80% | 80% |
| | Six-Features (optimized) | 87% | 96% |
| | Row Energy | 45% | 92% |
| “Distance” based library | Height/Width Ratio | 55% | 55% |
| | Six-Features (optimized) | 68% | 87% |
| | Row Energy | 86% | - |

5 Conclusions and Future Work

5.1 Summary

Persistent monitoring of the US border is an expensive effort in costs of manpower and equipment. Low-cost monitoring equipment is required to provide a cost-effective means of monitoring areas against illegal aliens and drug smugglers. Studies performed with sparse sensor array prototype at the Center for Advanced Sensors at the University of Memphis have shown that accurate classifications can be made of passing objects, reducing the rate of false detections caused by animals. This thesis presented further improvements of the sparse sensor array by describing the acquisition and real-time classification of an object on a low-resource microcontroller. A second prototype detector array, constructed of PVC pipe sections, allows the sensor to be disassembled and transported easily.

A microcontroller was utilized to combine the steps of data acquisition and profile classification onto a single, low cost device. This low resource platform solves the low power and low cost requirements for a UGS while minimizing false detection rates by discriminating between humans, vehicles, animals. Compressed, analyzed data is stored locally onboard the microcontroller. Hardware resources exist on the microcontroller (such as Ethernet, RS232, and RabbitNet) that would allow the microcontroller to communicate with an external network.

Better concealment of the sparse sensor array was achieved by removing the detectors of the array out of a single, obtrusive package. Removing the detectors from the single array, though, destroys the synchronization between the detectors. Algorithms were implemented to properly re-synchronize the timing between rows, allowing for

traditional feature extractions to be utilized for classification means. Additionally, the customized sensor array allows the variable of velocity to be calculated, further aiding classification. A feature extraction technique was introduced where classification results are independent of detector placement. Live testing of algorithms developed for the custom sensor array operating on the microcontroller was conducted, yielding real-time classification results of 85%. Subsequent post-processing of the data was capable of removing noise introduced in the real-time profile reconstruction technique. Post-processed profiles were classified according to their height to width feature and six-feature techniques using take-one-out studies, yielding 54% and 86% classification rates, respectively.

5.2 Works in progress / Future work:

Detectors of the custom sensor array are coupled with reflectors that are spaced at an equal horizontal distance from each other as their horizontally spaced detectors; beams from each detector are thus parallel to one another when viewed from overhead, as seen in Figure 5-1:

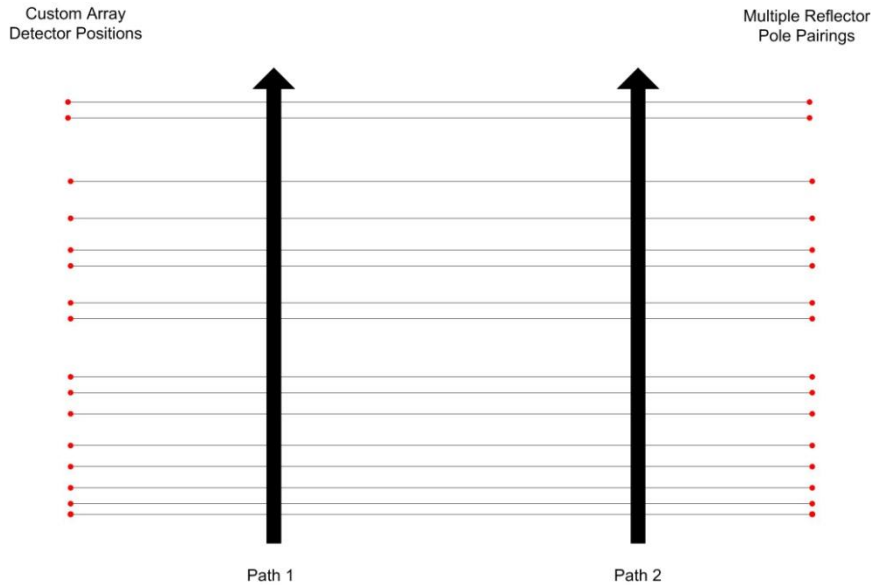


Figure 5-1. Custom Array with parallel beams and multiple reflector pole pairings, viewed from overhead.

Parallel beams present the requirements of additional hardware and setup. A subject passing through the sensor will have similar profiles independent of the distance he/she passed from the detectors or the paired reflector poles as illustrated in Figure 5-1. Information can be gathered from the subject profile, but the individual pathway is unknown, limiting potentially valuable information if a long distance exists between the detector array and its reflector pole. However, if the detector reflectors are positioned together along a single reflector pole, it is hypothesized that the distance from the sensor to the pathway of a passing subject can be calculated, as shown in Figure 5-2.

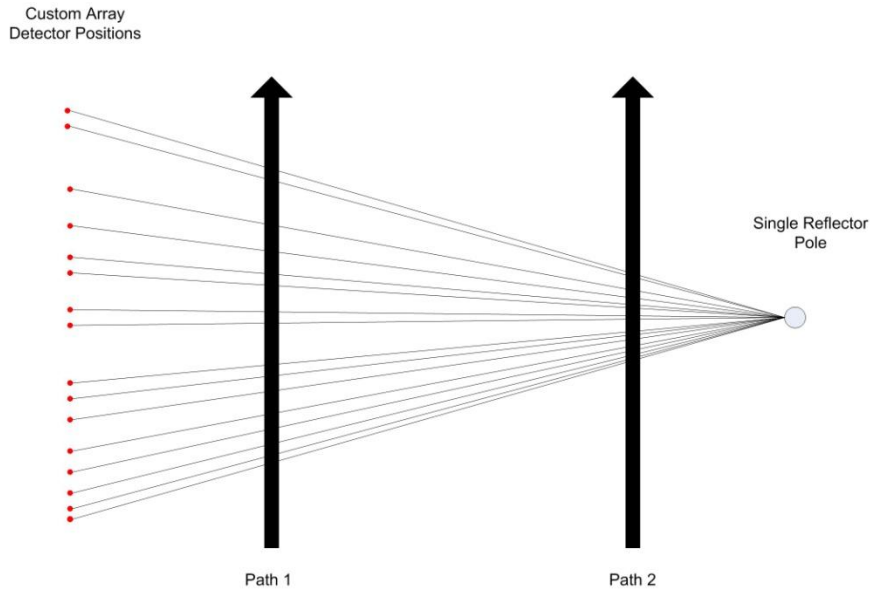


Figure 5-2. Custom Array with Single Reflector Pole.

A subject passing through the sensor at pathway 1 of Figure 5-2 will generate a wider profile than if he/she were to pass through pathway 2 at the same speed. A profile's width is now dependent both upon the subject's width and where it breaks the beams of the sensor's detectors, allowing us to infer a depth of field. However, it should also be noted that varying the subject's velocity can appear to vary its depth of field. For example, the profile of a fast-moving subject traveling on Figure 5-2's pathway 1 near the detector array would have a similar profile as if the same subject were traveling slowly on pathway 2. The speed or the depth of field can be calculated, but not both. However, by using the normalized row energy feature and classification technique described earlier, a reasonable estimate of speed can be made for the particular classification. By applying this reasonable speed to the profile, a depth of field can be determined.

Note that application of this non-parallel detector beam technique can also be implemented by separating the reflectors and placing the detectors together along the

same column, reducing the amount of necessary wiring from the detectors to the microcontroller. This depth of field calculation is not yet implemented. Maximum detector range of the CX-RVM5 sensing element limited to 5 meters; no important information is gained by knowing exactly where the subject passed through the 5 meter wide window other than it passed through. However, this field of view calculation will likely be beneficial once a longer-range detector/reflector pair is implemented into the detector array.

Additional future work involves exploitation of the microcontroller's Ethernet capabilities for data communication to an external network. Several supplemental software programs have been developed for the profile sensor to aid in sensor alignment. These supplemental programs are to be added to the data acquisition/classifier program to make a single, standalone, turnkey software package.

Future work also includes the collection of additional samples and rigorous testing of the custom array configuration against a more dynamic test subject set. Extensive hardware testing is to be performed to determine whether cooling and packaging considerations have been sufficiently addressed.

6 References

- [1] R.B. Sartain, "Profiling sensor for ISR applications," *Proceedings SPIE-Unattended Ground, Sea, and Air Sensor Technologies and Applications X*, volume 6963, pp. 69630Q, 2008.
- [2] D.J. Russomanno, C. Srikant, and C. Halford, "Sparse detector imaging sensor with two-class silhouette classification," *Sensors*, 8(12), 7996-8015, Dec. 2008.
- [3] D.J. Russomanno, M. Yeasin, E. Jacobs, M. Smith, and M.S. Sorrower, "Sparse detector sensor: profiling experiments for broad scale classification," *Proceedings SPIE-Defense and Security Symposium: Unattended Ground, Sea, and Air Sensor Technologies and Applications X*, volume 6963, pp. 69630M-69630M-11, Mar. 2008.
- [4] Ramco, "CX-RVM5/D100/ND300R datasheet," W. Des Moines, Ramco Innovations, 2007.
- [5] ACCES I/O Products, Inc."USB-DIO-32 USB Digital Input/Output Module," Retrieved 27 March 2011 from <http://accessio.com/go.cgi?p=../usb/usb-dio-32.html>.
- [6] Yeasin, M., Russomanno, D. J., Smith, M., Sorower, M. S., and Shaik, J., "Robust classification of objects from a sparse detection sensor," Proceedings of the International Conference on Machine Learning: Models, Technologies, and Applications, CSREA Press, Las Vegas, NV, 742-748 (2007).
- [7] K. Emmanuel, D. J. Russomanno, E. L. Jacobs, S. K. Chari, and J. B. Brown, "Silhouette Data Acquisition for a Sparse Detector Sensor," in *Proceedings of Military Sensing Symposia Passive Sensors*, SENSIAC, (2009).
- [8] Chari, S. K., Smith, F. A., Halford C. E., Jacobs, E. and Brooks, J., "Range and velocity independent classification of humans and animals using a profiling sensor," Proc. SPIE 7694, (2010).
- [9] Klett, K.K.; Sartain, R.; Alexander, T.; and Aliberti, K. Optical and Radiometry Analysis for a Passive Infrared Sparse Sensor Detection System. In *Proceedings SPIE-Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XIX* 2008; volume 6941, pp. 694101.
- [10] Brown, J. B., Chari, S. K. and Jacobs, E. L., "Assessment of a linear pyroelectric array sensor for profile classification," Proc. SPIE 7693. (2010).

- [11] Russomanno, D. J., Yeasin, M., Jacobs, E., Smith, M., and Sorower, E., "Sparse Detector Sensor: Profiling Experiments for Broad-Scale Classification," Proc. SPIE 6963 (2008).
- [12] Y. Tritenko, D.J. Russomanno and Q. Qiu, "A web service interface for an unattended ground sparse sensor detector," Proceedings of the International Conference on Semantic Web and Web Services, pp. 204-209, 2008.
- [13] Y. Tritenko, D.J. Russomanno and Q. Qiu, "Managing Sensor Deployments with Geographical Information Systems," *IEEE Sensors Applications Symposium*, IEEE Press, New Orleans, Louisiana, pp. 118-123, 2009.
- [14] Pelican Products, Inc. "1660 Case," Retrieved 27 March, 2011 from http://www.pelican.com/cases_detail.php?Case=1660.
- [15] Rabbit Semiconductor, Inc. "Rabbit 4000 Microprocessor User's Manual," Retrieved 27 March 2011 from ftp://ftp1.digi.com/support/documentation/0190152_h.pdf.
- [16] Digi International, Inc. "Dynamic C User's Manual," Retrieved 27 March 2011 from ftp://ftp1.digi.com/support/documentation/019-0167_F.pdf.
- [17] Digi International, Inc. "BL4S200 C-Programmable Single-Board Computer with Networking User's Manual," Retrieved 27 March 2011 from ftp://ftp1.digi.com/support/documentation/019-0171_D.pdf
- [18] Theodoridis, S. and Koutroumbas, K., [Pattern Recognition], Academic Press (2008).
- [19] Oppenheim, A. V. and Schaffer, R. K., [Discrete-Time Signal Processing], 3rd edition. Prentice Hall (2010).

APPENDIX

A1. A CD-ROM containing the profiling sensor's source code and related work has been provided to the University of Memphis Department of Electrical and Computer Engineering. Table A-1 briefly describes the content of the directories. Code developed in C is to be run using the Dynamic C integrated development environment version 10.56 or later. Dynamic C can be acquired through Digi®'s website: <http://www.digi.com/products/wireless-wired-embedded-solutions/software-microprocessors-accessories/software/dynamicc.jsp#overview>

Table A-1

| | |
|---------------------------|--|
| PS_vert.zip | Dynamic C code of a profiling sensor implementation using a vertically arranged row of detectors. Six-feature classification routine implemented for classification. |
| Vertical_library.zip | Library of samples acquired through PC/USB data acquisition system with the vertically oriented profile sensor. Data acquisition rate is approximately 1kHz. |
| PS_detector_align.zip | Dynamic C code used to aid in alignment of detector beams during profile sensor setup. |
| PS_detector_placement.zip | Dynamic C program used to calculate the physical distances between individual detectors of a custom array. |
| PS_custom_array.zip | Dynamic C program of a profiling sensor using a custom array. Program acquires, processes, classifies, and stores data. Six-feature, Height/width ratio, and Row-energy classification routines are implemented. |
| Distance_library.zip | Library of samples acquired using the Rabbit BL4S200 microcontroller. Data acquisition rate is approximately 21 Hz. |
| Matlab_routines.zip | Various Matlab routines used to visualize binary and .txt based profiles into a 2-dimensional image. |