Wright State University

CORE Scholar

[Browse all Theses and Dissertations](#)

[Theses and Dissertations](#)

2008

An Algorithm for the Detection of Handguns in Terahertz Images

Andrew J. Lingg

Wright State University

Follow this and additional works at: https://corescholar.libraries.wright.edu/etd_all

Part of the [Electrical and Computer Engineering Commons](#)

Repository Citation

Lingg, Andrew J., "An Algorithm for the Detection of Handguns in Terahertz Images" (2008). *Browse all Theses and Dissertations*. 899.

https://corescholar.libraries.wright.edu/etd_all/899

This Thesis is brought to you for free and open access by the Theses and Dissertations at CORE Scholar. It has been accepted for inclusion in Browse all Theses and Dissertations by an authorized administrator of CORE Scholar. For more information, please contact library-corescholar@wright.edu.

AN ALGORITHM FOR THE DETECTION OF HANDGUNS IN TERAHERTZ IMAGES

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Engineering

by

ANDREW J. LINGG
B.S.E.E., Wright State University, 2007

2008
Wright State University

WRIGHT STATE UNIVERSITY
SCHOOL OF GRADUATE STUDIES

December 17, 2008

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Andrew J. Lingg ENTITLED An Algorithm for the Detection of Handguns in Terahertz Images BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science in Engineering.

Brian D. Rigling, Ph.D.
Thesis Director

Kefu Xue, Ph.D.
Department Chair

Committee on
Final Examination

Brian D. Rigling, Ph.D.

Fred D. Garber, Ph.D.

William E. Pierson, Ph.D.

Joseph F. Thomas, Jr., Ph.D.
Dean, School of Graduate Studies

ABSTRACT

**Lingg, Andrew J. M.S. Egr., Department of Electrical Engineering, 2008.
An Algorithm for the Detection of Handguns in Terahertz Images.**

This paper presents an algorithm for detecting handguns in terahertz images. Terahertz radiation is capable of penetrating certain materials which are opaque at optical wavelengths, such as clothing, without the harmful effects of ionizing radiation. The approach taken is to segment objects of interest and classify them based on shape. We use a modified version of an active contour algorithm found in the open literature. Modifications include: a pre-processing step that includes clutter filtering and seeding of an initial contour; and a post-processing step that removes clutter pixels from the segmentation. The features used for classification are moment-based and Fourier shape descriptors. Classification as handgun or non-handgun from these features is done via Fisher's linear discriminant.

CONTENTS

1	Introduction	1
1.1	Previous Work	3
1.2	Outline	3
2	Data Set	5
3	Image Segmentation	15
3.1	Initial Seed	15
3.2	Active Contour	17
3.2.1	Vector Field Computation	18
3.2.2	Contour Deformation	20
3.3	Segmentation Refinement	23
4	Feature Computation and Classification	25
4.1	Feature Computation	25
4.2	Classifier Training	28
4.3	Classifier Testing	29
5	Results	30
6	Conclusion	36
A	ROC curves under various operating conditions	37
	References	51

LIST OF FIGURES

1.1	Block Diagram of Algorithm	2
2.1	Example terahertz image of a semi-automatic handgun	7
2.2	Example terahertz image of a semi-automatic handgun	8
2.3	Example terahertz image of a revolver handgun	9
2.4	Example terahertz image of a semi-automatic handgun	10
2.5	Example terahertz image of an ID card	11
2.6	Example terahertz image of a cellular phone	12
2.7	Example terahertz image of an iPod	13
2.8	Example terahertz image of a Nintendo DS	14
3.1	Initial contour example	17
3.2	Crimmins filtered image	19
3.3	Contour deformations	22
3.4	Segmentation result	24
5.1	Performance on the training set under resolution changes with no noise addition	31
5.2	Performance on the full resolution training set under noise corruption	32
5.3	Performance on the test set under resolution changes with no noise added	33
5.4	Performance on the full resolution test set under noise corruption . .	34
A.1	ROC curves for the training set at varying resolution corrupted by noise from $N(0, .1)$	37

A.2	ROC curves for the test set at varying resolution corrupted by noise from $N(0, .1)$	38
A.3	ROC curves for the training set at varying resolution corrupted by noise from $N(0, .2)$	39
A.4	ROC curves for the test set at varying resolution corrupted by noise from $N(0, .2)$	40
A.5	ROC curves for the training set at varying resolution corrupted by noise from $N(0, .4)$	41
A.6	ROC curves for the test set at varying resolution corrupted by noise from $N(0, .4)$	42
A.7	ROC curves for the training set at varying resolution corrupted by noise from $N(0, .8)$	43
A.8	ROC curves for the test set at varying resolution corrupted by noise from $N(0, .8)$	44
A.9	ROC curves for the training set at varying resolution corrupted by noise from $N(0, 1.6)$	45
A.10	ROC curves for the test set at varying resolution corrupted by noise from $N(0, 1.6)$	46
A.11	ROC curves for the training set at varying resolution corrupted by noise from $N(0, 3.2)$	47
A.12	ROC curves for the test set at varying resolution corrupted by noise from $N(0, 3.2)$	48
A.13	ROC curves for the training set at varying resolution across all noise levels	49
A.14	ROC curves for the test set at varying resolution across all noise levels	50

LIST OF TABLES

3.1	Initial Contour Seeding Algorithm	16
3.2	Table of Active Contour Parameters	21
3.3	Segmentation Refinement Algorithm	23

ACKNOWLEDGEMENT

I would first like to thank my advisor Brian Rigling for the outstanding support that he provided throughout this research. Thanks also to Fred Garber of WSU; Ed Zelnio, Erik Blasch, Greg Arnold, and Lee Patton of the Air Force Research Laboratory's Sensors Directorate; and Bill Pierson of iCAD Inc for their invaluable technical inputs. Finally, I would like to give an additional round of thanks to the members of my thesis committee: Brian Rigling, Fred Garber, and Bill Pierson

CHAPTER 1

INTRODUCTION

In many applications, it is desirable to be able to detect concealed weapons remotely and automatically[1]. In many security screening scenarios, manual searches and metal detectors are used to find dangerous objects and prevent them from being carried into a controlled area. However, metal detectors provide no warning against nonmetallic objects, and manual searches place security personnel at risk. Imaging technologies exploiting the millimeter wave and terahertz spectra can search for suspicious objects that may be concealed by clothing while avoiding the health hazards of ionizing radiation[2]. Automatic detection can also increase the throughput of a security checkpoint by expediting decisions or enabling management of several parallel screening points by fewer personnel.

In this paper, we present an algorithm designed to detect handguns in images from an active terahertz imager in the spatial domain. In our algorithm, a terahertz image, like the one shown in Figure 2.1, is segmented into target and clutter pixels using the active contour algorithm presented by Li and Acton [3], modified by preprocessing and postprocessing routines. A handgun/not-handgun decision is made based on the shape of the target region. Shape-based classification is commonly used in weapon detection algorithms operating on images. The algorithm in this paper is unique in its segmentation routine. In our segmentation algorithm, we first find a coarse initial segmentation via thresholding, refine via an active contour segmentation algorithm. A block diagram of this algorithm is shown in figure 1.1.

Figure 1.1: Block Diagram of Algorithm

1.1 Previous Work

Previous attempts at automatic detection have used either a spectral approach or a spatial approach. Spectral approaches to concealed weapon detection analyze the spectrum of the radar signature of screened individuals to determine if weapons are present. For example, Novak et. al. [4] present a system in which matched filters are used to discriminate between the spectral signatures of weapons and confuser objects. Spatial approaches utilize images formed from sensor data. Detection within these images occurs via image processing, computer vision, and pattern recognition techniques. Systems using a spatial approach first segment regions of interest from images, compute features from the segmented image, and classify the image based on these features. Object recognition via shape-based features has been used in many concealed weapon detection schemes [1, 5]. Several different approaches to image segmentation have been applied to this problem. The algorithm presented by Shen et. al. [6] segments passive terahertz images. It assumes that pixel intensities can be modeled by a mixture of Gaussian densities. Pixels are marked as target pixels if they are surrounded by the bright region that corresponds to the human in the image and the posterior probability of that pixel's intensity occurring is greatest for the density corresponding to concealed objects. Slamani et. al. [5] use the algorithm described in [7] to segment regions of interest. This algorithm separates regions by distinguishing between estimates of their probability distributions.

1.2 Outline

The remainder of this paper is organized as follows. Chapter 2 describes the data used to train and test the algorithm. Chapter 3 presents the seeded active contour segmentation algorithm used to find regions of interest. Features used for classification and the classification scheme are discussed in chapter 4. In chapter 5, we show the

results of our algorithm to detect handguns in the training and test data sets.

In this paper, we will use the following notation. Capital letters (e.g., X) represent matrices. Underlined lowercase letters (e.g., $\underline{x}$) indicate vectors, and lower case letters (e.g., x) represent scalars.

CHAPTER 2

DATA SET

The data operated on are real-valued intensity images of varying sizes. These images were formed from a terahertz imager developed by Petkie et. al. [8]. This imager has a center frequency of $f_c = 640$ GHz. In this imager, a directed beam of terahertz radiation is aimed at a mirror. This mirror is rotated by a set of servo motors such that the reflected beam is raster scanned across the scene. For each pixel in the image, the pitch and yaw angles of the servos are encoded into pixel coordinates, and the amplitude of the radiation reflected by the scene is recorded as the pixel amplitude. All of the images used were captured with the target at a distance of approximately 1 meter from the imager.

The available data consists of a training set of 27 images, 12 containing handguns and 15 containing non-threat objects, and a test set containing 7 handguns and 17 non-threat objects that are withheld from training. Figures 2.1 through 2.8 are examples of the image data used.

Of the 12 images containing handguns in the training set, 7 were semi-automatics, and 5 were revolvers. Of the five images containing revolvers, 3 were captured with no obstruction between the imager and the target, and 2 were captured with a brown cotton sheet covering the target. Six of the seven images containing semi-automatic handguns were captured with the broadside of the target at approximately normal incidence to the imager. Four of these images were taken with no obstruction between the imager and the target; one was taken with the target covered by a brown cotton sheet; and two were taken with the target covered by a thick blanket. The seventh

image was captured with the broadside of the target at approximately 30 degrees from normal incidence and covered by a brown cotton sheet.

The images of non-threat objects in the training set were all captured with the broadside of the object at approximately normal incidence to the imager. These images contained the following objects: 1 image of a D-cell battery covered by a cotton sheet; 1 image of a scientific calculator; 1 image of a set of calipers; 1 image of a cellular telephone covered by a cotton sheet; 1 image of a pack of cigarettes; 3 images of a Nintendo DS, 1 of the back side covered by a cotton sheet, 1 of the front side covered by a cotton sheet, and 1 of the front side with no obstruction; 1 image of an ID card covered by a cotton sheet; and 6 images of an iPod mini, back side both covered by a cotton sheet and without cover, back side with earphones wrapped around the device both covered by a cotton sheet and without cover, and front side with earphones wrapped around the device both covered by a cotton sheet and without cover.

All 7 images of handguns in the test set contained semi-automatics. Five were taken at approximately normal incidence to the imager. Of the 5 captured at normal incidence, 2 were covered by a cotton sheet, 1 by the blanket, and 2 were captured with no obstructions. The last 2 images were captured with the broadside of the target at 10 and 20 degrees from normal incidence, and were covered by a cotton sheet.

The images of non-threat objects in the test set contained: 1 image of an aluminum soda can covered by a cotton sheet; 2 images containing different types of cellular phones; 1 image of a pack of cigarettes covered by a cotton sheet; 1 image of a flashlight covered by a cotton sheet; 1 image of the back side of a Nintendo DS with no cover; 1 image of the front side of an iPod mini covered by a cotton sheet; 1 image of a short length of metal pipe; 2 images of a short length of PVC tube; 2 images of a short length of PVC tube covered by a cotton sheet; 2 images of a 2-way radio, 1 covered by a cotton sheet and 1 with no cover; and 2 images of a triangular ruler, 1

covered by a cotton sheet and 1 covered by a thick blanket.

Figure 2.1: Example terahertz image of a semi-automatic handgun

Figure 2.2: Example terahertz image of a semi-automatic handgun

Figure 2.3: Example terahertz image of a revolver handgun

Figure 2.4: Example terahertz image of a semi-automatic handgun

Figure 2.5: Example terahertz image of an ID card

Figure 2.6: Example terahertz image of a cellular phone

Figure 2.7: Example terahertz image of an iPod

Figure 2.8: Example terahertz image of a Nintendo DS

CHAPTER 3

IMAGE SEGMENTATION

In segmenting an image, we attempt to determine which regions of the image contain objects of interest and which regions contain clutter. Features computed from the segmented image are used for classification. Many algorithms for image segmentation were tested, such as thresholding methods, edge detection methods, region growing[9], and active contours[3, 10, 11]; however, all are known to have problems operating under certain conditions. The best results were obtained by modifying the active contour algorithm presented by Li and Acton[3] to include an initial seeding process prior to contour deformation and a refinement process on the final contour. Because the behavior of active contours are data driven, provisions for an error in contour deformation, such as contours moving off of the image edge, have also been implemented.

3.1 Initial Seed

Active contour methods are well suited to localizing object boundaries. They are based on minimization of an energy function that will reach a minima when the contour finds an edge in the image. However, if initialized far from objects of interest in background regions containing strong clutter, an active contour may instead conform to a local minima that is not a true object boundary. A method of dealing with this problem is to seed an initial contour close to the true object boundary. For this application, thresholding methods based on image statistics have

Table 3.1: Initial Contour Seeding Algorithm

1	Erosion; Disk, radius 10
2	Threshold; Choose pixels brighter than $\mu + .5\sigma$
3	Remove small objects; Only consider objects > 5000 pixels
4	Label Regions
5	Keep Region Containing Brightest Pixel
6	Binary Erosion; Disk, radius 5
7	Label Regions
8	Keep Region Containing Brightest Pixel
9	Compute Convex Hull
10	Binary Dilation; 3x3 square, 24 times

proven capable of finding the “general area” of the image containing the object.

We seed the initial contour so that it fully contains a large region of bright pixels, as such a region is likely to be an object of interest. Table 3.1 lists the steps taken to seed the initial contour from full resolution images. Note that object sizes, structuring element sizes, and resampling factors are scale and resolution dependent. The values listed here are the values used while operating on our data set.

Step 1 reduces the influence of bright clutter pixels. Step 2 marks bright pixels, as we know that objects of interest will be brighter than the background. Step 3 removes small clutter regions that are considered by step 2. Steps 4 and 5 remove all regions except the one containing the brightest pixel. Steps 6, 7, and 8 break any weakly connected regions from the segmentation, keeping only the region with the brightest pixel. Steps 9 and 10 remove concavities from and expand the segmented region, likely resulting in a region in which the object is completely contained. The result of this algorithm is a binary image. The region seeded from the example image in Figure 2.1 is shown in Figure 3.1 with the black line representing the boundary of the “on” pixels of the binary image. We choose a region that is likely to be larger than the object because the contour will shrink during deformation (section 3.2.2) when not near features of interest. The resulting segmentation will be downsampled by a factor of 8 in each dimension in order to reduce computation time of the contour

evolution, and is used as an input to the active contour algorithm. Occasionally, an error occurs during the contour evolution when the contour moves off of the image edge. This error occurred once among the 54 images at full resolution with no added noise corruption. Should this error occur, a copy of the segmentation is saved between steps 8 and 9, is downsampled to match the size of the active contour output, and sent to the segmentation refinement algorithm.

Figure 3.1: Initial contour example

3.2 Active Contour

The inputs given to the active contour are the original image, decimated by a factor of 8 in each dimension, and the result of pre-segmentation. Here, we employ the vector field presented by Li and Acton[3] and use the implementation of contour

deformation given by Xu and Prince[11, 12], which uses the method given by Kass et al [10]. We initialize the contour to the boundary resulting from the seeding algorithm. The active contour algorithm first generates a vector field based on an edge image, then deforms based on the minimization of an energy function in equation (3.7). We discuss this process in detail below, highlighting modifications for use with our data set.

3.2.1 Vector Field Computation

Computing the vector field is the first step in the active contour algorithm. This vector field is computed by convolving a vector field kernel with an edge map of the image. Rather than compute an edge map from the original image, we attempt to suppress clutter and noise using ten iterations of the Crimmins speckle removal filter[13]. The Crimmins filter operates on a quantized image, in this case quantized to integers on $[0, 255]$. This filter smooths an image by increasing the intensity of pixels at local minima and decreasing the intensity of pixels at local maxima. The result of the Crimmins filtering is shown in Figure 3.2.

We then compute an edge image, $I(x, y)$, by convolving the output of the Crimmins filter with a Laplacian of Gaussian filter

$$I(x, y) = F(x, y) * \nabla^2 G(x, y) \quad (3.1)$$

where, for our data set, we define the 13×13 filter

$$\nabla^2 G(x, y) = \frac{-1}{\pi\sigma^4} \left(1 - \frac{x^2 + y^2}{2\sigma^2}\right) e^{\frac{-(x^2 + y^2)}{2\sigma^2}}, \quad (3.2)$$

with $\sigma = 2$, $x = [-6, \dots, 6]$ and $y = [-6, \dots, 6]$. Formation of the edge image is followed by truncation of pixels below the 1st and above the 99th percentile of pixel intensities, compensating for outlier effects. The pixel intensities of the edge image

Figure 3.2: Crimmins filtered image

are linearly scaled and shifted onto the interval $[-\frac{1}{2}, \frac{1}{2}]$. Pixels in the scaled image with an absolute value less than 0.1 are zeroed as probable noise-induced edges.

The vector field is computed by convolution

$$K(x, y) = I(x, y) * C(x, y) \quad (3.3)$$

with the kernel

$$C(x, y) = M(x, y)N(x, y) \quad (3.4)$$

where $M(x, y) = e^{\frac{-r^2}{\zeta^2}}$ is the magnitude of the kernel, and $N(x, y) = \frac{-x}{r} + j\frac{-y}{r}$ is a complex-valued array whose vectors point to the origin of the kernel. Where $r = \sqrt{x^2 + y^2}$, $x = [-25, -24, \dots, 25]$, $y = [-25, -24, \dots, 25]$, $\zeta = 2.5$, and $N(0, 0) = 0$.

We extract the horizontal and vertical components of the vector field by

$$U(x, y) = \Re\{K(x, y)\} \quad (3.5)$$

$$V(x, y) = \Im\{K(x, y)\} \quad (3.6)$$

where $\Re\{.\}$ and $\Im\{.\}$ denote taking the real and imaginary parts respectively.

3.2.2 Contour Deformation

We initialize the contour of the region of interest to be the boundary of the binary image obtained from the seeding algorithm in section 3.1. This contour deforms based on influences from the vector field as well as the smoothness and stiffness of the contour itself.

We first develop a continuous-domain representation of the energy function governing the contour. Let $\underline{z}(s) = (x(s), y(s))$, $s \in [0, 1]$ be a parametric curve representing

Table 3.2: Table of Active Contour Parameters

Parameter	Meaning	Value
α	Elasticity	$1.5(1 + e^{-.5(N-50)})^{-1} + .5$
β	Rigidity	0.1
γ	Viscosity	1
κ	Time Step	1
N	A counting variable	$0.2 \times \text{Current iteration number}$

the contour. We seek a contour that minimizes the energy function (see [3])

$$e_{ac} = \int_0^1 \left[\frac{1}{2} \left(\alpha \left| \frac{dz(s)}{ds} \right|^2 + \beta \left| \frac{d^2 z(s)}{ds^2} \right|^2 \right) - |K(z(s))|^2 \right] ds, \quad (3.7)$$

where $K(\cdot)$ is the continuous-domain version of the vector field described by (3.3).

In practice, we must find this contour iteratively. First, we define the parameters of the active contour (defined in Table 3.2) and the values used for them in this application. Let A be an $m \times m$ Toeplitz matrix where

$$A(k, l) = \begin{cases} 2\alpha + 6\beta, & k = l \\ -\alpha - 4\beta, & |k - l| = 1 \\ \beta, & |k - l| = 2 \\ 0, & \text{otherwise} \end{cases} \quad (3.8)$$

From A , we compute the matrix

$$B = (A + \gamma I)^{-1} \quad (3.9)$$

where I is the identity matrix. If the points on the contour at time t are $\underline{x}(t)$ and

$\underline{y}(t)$, then they are updated at each iteration by

$$\underline{x}(t+1) = B(\gamma \underline{x}(t) + \kappa \underline{u}(\underline{x}, \underline{y})) \quad (3.10)$$

$$\underline{y}(t+1) = B(\gamma \underline{y}(t) + \kappa \underline{v}(\underline{x}, \underline{y})), \quad (3.11)$$

where $\underline{u}$ and $\underline{v}$ are $m \times 1$ vectors containing the values stored in U and V at the coordinates pairs stored in $\underline{x}$ and $\underline{y}$.

For this application, the contour deforms for 1000 iterations, and every 5 iterations, the contour is interpolated so that each point is a distance of one pixel from the previous. Figure 3.3 shows the progression of a contour. The gray contours are intermediate forms, and the white contour is the final result.

Figure 3.3: Contour deformations

Table 3.3: Segmentation Refinement Algorithm

1	Resize copy of original image to segmentation size
2	Keep pixels in segmentation greater than the mean of those pixels or in the 99th percentile of the image
3	Remove small objects; Only consider objects > 3 pixels
4	Binary Dilation; Disk, radius 5
5	Fill Holes
6	Binary Erosion; Disk, radius 5
7	Keep Largest Object
8	Turn on pixels for which another pixel is turned on between it and the image edge in 3 or more directions out of 4(NSEW).
9	Take intersection of the result with that produced by the active contour
10	Keep Largest Object
11	Fill Holes

3.3 Segmentation Refinement

For most of the images in the data set, the seeded active contour will produce a satisfactory segmentation. Most often, if a segmentation is not satisfactory, the problem is that it contains a clutter region adjacent to the object. The purpose of the segmentation refinement algorithm is to remove small clutter regions that were included in the segmentation. It also contains provisions for recovering target pixels lost in the clutter removal step.

This problem is addressed by the segmentation refinement algorithm, outlined in Table 3.3. Steps 1, 2, and 3 remove pixels that are likely clutter. Steps 4 and 5 rebuild any strong connections, lost in steps 1-3, between segmented target regions. Step 6 undoes the effect of step 4 on weakly connected regions. Step 7 removes small regions disconnected from the segmentation. Step 8 fills concavities in the segmentation formed by the removal of darker target pixels while steps 9 and 10 prevent unreasonable results from step 8 from becoming part of the segmentation.

Finally, step 11 ensures a solid segmentation. The final segmentation result on the example image from Figure 2.1 is shown in Figure 3.4.

Figure 3.4: Segmentation result

CHAPTER 4

FEATURE COMPUTATION AND CLASSIFICATION

From the segmented image, we compute a set of shape based features. After features have been computed, a class decision must be made based on these features. Here, we consider only two classes: handgun not present and handgun present, hereafter referred to as H_0 and H_1 respectively. We use Fisher's Linear Discriminant[14] to classify the feature vectors generated from each image. This classifier was chosen because it only requires estimates of class means and covariances, rather than a complete knowledge of the underlying distributions of features, and is simple to understand and implement. Like many classifiers, Fisher's Linear Discriminant has a training phase and an operation phase. Each of these will be discussed in the following.

4.1 Feature Computation

The features used for classification are dependent on the shape of the object, while invariant to translation, rotation, and scale. These shape descriptors are computed from the binary image output by the segmentation algorithm. All 8 of these descriptors can be found in [1]. The first six of these are the moment based shape descriptors

presented by Hu:[15].

$$d_1 = \eta_{20} + \eta_{02} \quad (4.1)$$

$$d_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \quad (4.2)$$

$$d_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad (4.3)$$

$$d_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \quad (4.4)$$

$$\begin{aligned} d_5 = & (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ & + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})] \end{aligned} \quad (4.5)$$

$$d_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \quad (4.6)$$

where η_{pq} is the normalized central moment and μ_{pq} is the central moment:

$$\eta_{pq} = \frac{\mu_{pq}}{\frac{p+q+2}{2} \mu_{00}} \quad (4.7)$$

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q \quad (4.8)$$

The features are based on the uniqueness theorem concerning moments that is also presented by Hu[15]. This theorem states that a probability density function can be uniquely determined by the sequence of its moments. Using normalized central moments removes dependence on translation and scale. The descriptors combine the moments in such a way to remove dependence on rotation while preserving sensitivity to changes in shape.

The remaining two features are Fourier shape descriptors. Zahn and Roskies[16] present the Fourier descriptors and empirically demonstrate their utility in shape recognition. Recall that $\underline{x}$ and $\underline{y}$ are $m \times 1$ vectors containing the coordinates of each point on the object contour. Re-writing these vectors as sequences of length m , $x(i)$

and $y(i)$, we define the complex-valued sequence

$$z(i) = x(i) + jy(i), \quad (4.9)$$

which is periodic for each complete traversal of the boundary. We then take the discrete Fourier transform of the boundary.

$$Z(k) = \frac{1}{n} \sum_{i=0}^{n-1} z(i) e^{-j \frac{2\pi k i}{n}} \quad (4.10)$$

The terms in the sequence $Z(k)$ are known as the Fourier descriptors. In order to remove dependence on position, orientation, and scale, we set the DC term of $Z(0) = 0$ and normalize by the rest of the sequence by the first AC term.

$$\hat{Z}(k) = \frac{Z(k)}{Z(1)} \quad (4.11)$$

Where $\hat{Z}(k)$ is the sequence known as the normalized Fourier descriptors. We compute the shape descriptors used for classification from the normalized Fourier descriptors by

$$d_7 = \sum_{k=2}^{n-1} |\hat{Z}(k)| \quad (4.12)$$

$$d_8 = \begin{cases} \sum_{k=2}^{\frac{n}{2}} |\hat{Z}(k) + \hat{Z}(n-k+2)| & \text{n is even} \\ \sum_{k=2}^{\frac{n-1}{2}} |\hat{Z}(k) + \hat{Z}(n-k+2)| & \text{n is odd} \end{cases} \quad (4.13)$$

The quantities d_1 through d_8 form a feature vector on which a classifier operates.

4.2 Classifier Training

The purpose classifier training is to set a decision boundary using sample data from each class. The decision boundary drawn by Fisher's linear discriminant is a hyperplane of dimension $d - 1$ where d is the number of features. In this case, we will generate a hyperplane in $\Re_8$, and the class decision is made based on which side this hyperplane a feature vector lies. This is equivalent to projecting data onto a line that is orthogonal to the hyperplane and checking the position of the data point on that line relative to the point that the line crosses the hyperplane.

In training, we will find this line directly such that the ratio of the difference in class means to the within class covariance is maximized. We find a vector along this line by the optimization [14]

$$\underline{w} = \arg \max_{\underline{\hat{w}}} \left\{ \frac{|\underline{\hat{w}}^T(\underline{m}_1 - \underline{m}_0)|^2}{\underline{\hat{w}}^T S_w \underline{\hat{w}}} \right\} \quad (4.14)$$

with solution

$$\underline{w} = S_w^{-1}(\underline{m}_1 - \underline{m}_0) \quad (4.15)$$

where $\underline{m}_1$ and $\underline{m}_0$ are the means of the data points from H_1 and H_0 . The variable S_w is the pooled within-class covariance matrix of the the sampled set, given by

$$S_w = \frac{1}{n - 2}(n_1 \hat{\Sigma}_1 + n_0 \hat{\Sigma}_0) \quad (4.16)$$

where n is the number of sample points, n_1 and n_0 are the number of sample points from H_1 and H_0 , and $\hat{\Sigma}_1$ and $\hat{\Sigma}_0$ are estimates of the covariance matrices of each class. The numerator of (4.14) represents the separation between the class means after projection onto $\underline{w}$ while the denominator represents the within-class covariance

after projection onto $\underline{w}$.

4.3 Classifier Testing

Results from training used in operation are the class means and the vector $\underline{w}$. When a new feature vector $\underline{p}$ is to be classified, it is mean adjusted and projected onto $\underline{w}$ to give the detection statistic $h(\underline{p}) = (\underline{p} - \frac{1}{2}(\underline{m}_1 + \underline{m}_2))^T \underline{w}$. This projection is compared to a threshold to make a class decision such that

$$c(\underline{p}) = \begin{cases} H_0, h(\underline{p}) \leq t \\ H_1, h(\underline{p}) > t \end{cases} \quad (4.17)$$

is the class to which $\underline{p}$ belongs. Most values suggested for t in literature are dependent on the prior probability that a given sample will belong to a certain class [14]. However, here the priors are not known, so we instead choose a threshold based on the desired operating point on the ROC curve (see Section 5).

CHAPTER 5

RESULTS

Using the training and test data described in section 2, we assess the performance of this algorithm against terahertz images containing handguns, cellular phones, handheld gaming devices, music players, and several other items that may be carried by travelers or by workers entering a controlled site. We evaluate the algorithm’s performance on the test and training sets individually. Additionally we assess the algorithm’s performance on each of these data sets as a function of resolution, noise, rotation, and scaling. To simulate a decrease in resolution, we decimate the images. To simulate noise corruption, we add zero-mean white Gaussian noise to the images. To demonstrate invariance to rotation and scale, the binary images produced by the segmentation algorithm are rotated and scaled during testing. The amount of rotation is randomly selected from a uniform density containing all multiples of 5 between 0 and 355 degrees. The amount of scaling is randomly selected from a uniform density containing the powers of 2 from 2^{-2} to 2^2 . Ten realizations of rotation, scaling, and noise corruption are tested for each image at each resolution/noise power combination.

Figures 5.1 and 5.2 show the ROC curves for the algorithm operating on the training set under changes in resolution and noise levels respectively, and Figures 5.3 and 5.4 show the ROC curves for the algorithm operating on the test set under changes in resolution and noise levels. Performance between training and test data is similar, indicating good generalization. As expected, the algorithm’s performance declines as resolution is decreased and noise is added. The decrease in performance

Figure 5.1: Performance on the training set under resolution changes with no noise addition

Figure 5.2: Performance on the full resolution training set under noise corruption

Figure 5.3: Performance on the test set under resolution changes with no noise added

Figure 5.4: Performance on the full resolution test set under noise corruption

between full resolution data and resolution spoiled by 2 is much smaller than the difference between data spoiled by 2 and spoiled by 4. With resolution spoiled by 4, the segmentation algorithm begins to break down, resulting poorer overall performance. Performance under additional combinations of resolution and noise corruption are shown in Appendix A.

CHAPTER 6

CONCLUSION

We have developed an algorithm for detecting handguns in terahertz images. This algorithm uses a seeded active contour to segment an object of interest and classifies it based on shape. Our algorithm is effective for analyzing images from our data set containing a single object imaged with its broadside at near-normal incidence to the imager.

Further work is needed to extend the initial contour seeding algorithm to the multiple object case. A contour could then be evolved on each individual object to obtain an accurate segmentation. The features used for classification are invariant to similarity transformations (translation, rotation, and scale) but not affine transformations. Affine transformations of the 2-D projection onto the image will occur if the object of interest is rotated about an axis other than one orthogonal to the image plane. Use of features that are invariant to affine transformations will allow detection of objects that have been rotated out of the image plane. Finally, methods for recognizing partially occluded objects will be necessary in order to be effective in a practical deployment.

APPENDIX A

ROC CURVES UNDER VARIOUS OPERATING CONDITIONS

Figure A.1: ROC curves for the training set at varying resolution corrupted by noise from $N(0,1)$

Figure A.2: ROC curves for the test set at varying resolution corrupted by noise from $N(0,1)$

Figure A.3: ROC curves for the training set at varying resolution corrupted by noise from $N(0, .2)$

Figure A.4: ROC curves for the test set at varying resolution corrupted by noise from $N(0, .2)$

Figure A.5: ROC curves for the training set at varying resolution corrupted by noise from $N(0, .4)$

Figure A.6: ROC curves for the test set at varying resolution corrupted by noise from $N(0, .4)$

Figure A.7: ROC curves for the training set at varying resolution corrupted by noise from $N(0, .8)$

Figure A.8: ROC curves for the test set at varying resolution corrupted by noise from $N(0,.8)$

Figure A.9: ROC curves for the training set at varying resolution corrupted by noise from $N(0,1.6)$

Figure A.10: ROC curves for the test set at varying resolution corrupted by noise from $N(0, 1.6)$

Figure A.11: ROC curves for the training set at varying resolution corrupted by noise from $N(0, 3.2)$

Figure A.12: ROC curves for the test set at varying resolution corrupted by noise from $N(0, 3.2)$

Figure A.13: ROC curves for the training set at varying resolution across all noise levels

Figure A.14: ROC curves for the test set at varying resolution across all noise levels

REFERENCES

- [1] H.-M. Chen, S. Lee, R. M. Rao, M. A. Slamani, and P. K. Varshney, “Imaging for concealed weapon detection: a tutorial overview of development in imaging sensors and processing,” *IEEE Signal Processing Magazine*, vol. 22, no. 2, pp. 52–61, Mar. 2005.
- [2] D. Brenner, R. Doll, D. Goodhead, E. Hall, C. Land, J. Little, J. Lubin, D. Preston, R. Preston, J. Puskin *et al.*, “Cancer risks attributable to low doses of ionizing radiation: Assessing what we really know,” *Proceedings of the National Academy of Sciences*, vol. 100, no. 24, pp. 13 761–13 766, 2003.
- [3] B. Li and S. T. Acton, “Active contour external force using vector field convolution for image segmentation,” *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2096–2106, Aug. 2007.
- [4] D. Novak, R. Waterhouse, and A. Farnham, “Millimeter-wave weapons detection system,” in *Applied Imagery and Pattern Recognition Workshop, 2005. Proceedings. 34th*, Oct. 2005.
- [5] M. A. Slamani, P. K. Varshney, R. M. Rao, M. G. Alford, and D. Ferris, “Image processing tools for the enhancement of concealed weapon detection,” in *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, vol. 3, Kobe, Oct. 1999, pp. 518–522.
- [6] X. Shen, C. R. Dietlein, E. Grossman, Z. Popovic, and F. G. Meyer, “Detection and segmentation of concealed objects in terahertz images,” *IEEE Transactions on Image Processing*, vol. 17, no. 12, pp. 2465–2475, December 2008.
- [7] M. A. Slamani, D. D. Weiner, and V. Vannicola, “A new statistical procedure for the segmentation of contiguous nonhomogeneous regions based on the ozturk algorithm,” in *Proc. SPIE Conf. Statistical and Stochastic Methods for Image Processing*, 1996.
- [8] D. Petkie, F. De Lucia, C. Castro, P. Helminger, E. Jacobs, S. Moyer, S. Murrill, C. Halford, S. Griffin, and C. Franck, “Active and passive millimeter-and sub-millimeter-wave imaging,” in *Proceedings of SPIE*, vol. 5989. SPIE, 2005, p. 598918.
- [9] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*. Cole Publishing Company, 1996.

- [10] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, January 1988.
- [11] C. Xu and J. L. Prince, “Snakes, shapes, and gradient vector flow,” *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 359–369, Mar. 1998.
- [12] ———, “Active contours, deformable models, and gradient vector flow,” 2002–2008, <http://www.iac1.ece.jhu.edu/projects/gvf/>. [Online]. Available: <http://www.iac1.ece.jhu.edu/projects/gvf/>
- [13] T. Crimmins, “The geometric filter for speckle reduction,” *Applied Optics*, vol. 24, no. 10, May 1985.
- [14] A. Webb, *Statistical Pattern Recognition*, 2nd ed. John Wiley & Sons, Ltd, 2002.
- [15] M.-K. Hu, “Visual pattern recognition by moment invariants,” *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, Feb. 1962.
- [16] C. T. Zahn and R. Z. Roskies, “Fourier descriptors for plane closed curves,” *IEEE Transactions on Computers*, vol. C-21, pp. 269–281, 1972.