E-SHAPE ANALYSIS

Paul Sroufe, BSCE

Thesis Prepared for the Degree of

MASTER OF SCIENCE

UNIVERSITY OF NORTH TEXAS

December 2009

APPROVED:

Ram Dantu, Major Professor
João Cangussu, Committee Member
Phil Sweany, Committee Member
Bill Buckles, Program Coordinator
Ian Parberry, Chair of the Department of Computer Science and Engineering
Costas Tsatsoulis, Dean of the College of Engineering
Michael Monticino, Dean of the Robert B. Toulouse School of Graduate Studies

Sroufe, Paul. E-Shape Analysis. Master of Science (Computer Science and Engineering), December 2009, 49 pp., 13 tables, 28 illustrations, 21 references.

The motivation of this work is to understand E-shape analysis and how it can be applied to various classification tasks. It has a powerful feature to not only look at what information is contained, but rather how that information looks. This new technique gives E-shape analysis the ability to be language independent and to some extent size independent.

In this thesis, I present a new mechanism to characterize an email without using content or context called E-shape analysis for email. I explore the applications of the email shape by carrying out a case study; botnet detection and two possible applications: spam filtering and social-context based finger printing.

The second part of this thesis takes what I apply E-shape analysis to activity recognition of humans. Using the Android platform and a T-Mobile G1 phone I collect data from the triaxial accelerometer and use it to classify the motion behavior of a subject.

ACKNOWLEDGMENTS

This work is supported by the National Science Foundation under grants CNS-0627754, CNS-0619871 and CNS-0551694.

I would like to acknowledge the following people as they have supported me through my masters: Dr. Ram Dantu, Dr. João Cangussu, Dr. Santi Phithakkitnukoon, and Dr. Phil Sweany.

CONTENTS

ACKNOWLEDGEMENTS iii

LIST OF TABLES v

LIST OF FIGURES vi

CHAPTER 1. INTRODUCTION 1

1.1. E-Shape Concept 1

1.2. Previous Work Related to Botnet Detection 1

1.3. Applications of E-Shape 2

1.4. Main Contributions 3

1.5. G1 Experiments 3

CHAPTER 2. E-Shape ANALYSIS OF EMAIL 4

2.1. Introduction 4

2.2. Email Shape 5

2.3. Applications of E-Shape 7

2.3.1. Spam Botnet Detection 8

2.4. Future Work and Preliminary Results 12

2.4.1. Spam Filtering 13

2.4.2. Social Context-Based Finger Print Detection 14

2.5. Limitations of the Study 16

2.6. Conclusion 17

CHAPTER 3. G1 INFRASTRUCTURE AND ACCELEROMETER READINGS 18

3.1. Introduction 18

3.2. Previous Work in Activity Detection and Accelerometer Readings 18

3.3. Problem Definition 19

3.4. The Android G1 Platform 19

3.5. Experimental Data 20

3.5.1. Walking 21

3.5.2. Running 22

3.5.3. Upstairs 22

3.5.4. Downstairs 23

3.6. Experiments for Activity Detection 23

3.6.1. Experiment 1: Signal Correlation 24

3.6.2. Experiment 2: Shape Analysis for Activity Detection 27

3.6.3. Experiment 3: Signal Covariance 30

3.7. Conclusion 34

CHAPTER 4. DAILY BEHAVIORAL SHAPE 36

4.1. Introduction 36

4.2. G1 Daily Activity Analysis 36

4.2.1. Previous Work in Activity Recognition 36

4.2.2. Accelerometer Data Information and Collection 37

4.2.3. Feature Selection for Daily Activities 38

4.2.4. G1 Activity Recognition from Training Data 38

4.2.5. Python Classification Tool Using Python Orange 39

4.2.6. E-Shape Analysis as Applied to Daily Activity Recognition 40

4.3. G1 Individual Walking Analysis 43

4.3.1. E-Shape Analysis 44

CHAPTER 5. CONCLUSIONS 46

BIBLIOGRAPHY 48

LIST OF TABLES

2.1 Details of dataset for botnet detection experiment. 10

2.2 Accuracy of individual bucket. 13

3.1 Cross-correlation of a walking sample validated against the other samples.. . . . 26

3.2 Cross-correlation of a upstairs sample validated against the other samples. . . . 26

3.3 Mean Cross-covariance above 0; Running Sample 1.. 33

3.4 Mean Cross-covariance above 0; Walking Sample 1.. 33

3.5 Mean Cross-covariance above 0; Upstairs Sample 1.. 34

4.1 Confusion matrix for daily activities. 37

4.2 Total features with selection. 38

4.3 Classification of unknown data from training set. 39

4.4 Context to activity mappings. (Italicized items are for future work.) 41

4.5 Confusion Matrix for distance analysis of daily shapes. 43

4.6 Confusion matrix for individual walking analysis. 44

LIST OF FIGURES

2.1 Email shape analyzer. 6

2.2 An example of four different E-shapes. 7

2.3 An example of the actual spam botnet template. 10

2.4 Showing size independence of shapes from the same botnet. 11

2.5 A result of the botnet detection experiment based on 879 different size and language emails. 12

2.6 A result of the spam filtering experiment based on 7,500 emails from TREC 2007 corpus. 14

2.7 A result of the social context-based individual's personal email finger print experiment based on three different individual email senders to a subject with total of about 250 emails.. 15

3.1 Screenshot of blackbox GUI. 20

3.2 Walking signal for approximately forty-five seconds. 21

3.3 Running signal (Y-axis) for approximately forty-five seconds. 22

3.4 Discovery Park stairwell, walking upstairs. 22

3.5 Discovery Park stairwell, walking downstairs.. 23

3.6 Auto cross-correlation of the walking signal. 24

3.7 Correlation of 15sec of walking to total walking sample. 25

3.8 Correlation of 15sec of running to total running sample. 25

3.9 Walking Data 1, Hellinger analysis versus Walking Data 2. 29

3.10 Running data 1, Hellinger analysis versus Running Data 2. 29

3.11 Running sample versus walking sample; Hellinger analysis.. 29

3.12 Auto cross-covariance of 45sec walking data set. 30

3.13 Cross-covariance walking 15sec to total sample set. 31

3.14 Cross-covariance running 15sec to total sample set. 31

3.15 Cross-covariance walking to downstairs sample sets.. 32

3.16 Cross-covariance walking to upstairs sample sets. 32

4.1 Data flow diagram for accelerometer readings.. 37

4.2 Side by side of two Monday's raw accelerometer readings. 40

4.3 Side by side of two Tuesday's raw accelerometer readings. 41

4.4 Comparison of E-shapes for Monday 2 and 3. 42

4.5 Comparison of E-shapes for Tuesday 2 and 3. 42

4.6 Walking Signal for approximately forty-five seconds.. 45

CHAPTER 1

INTRODUCTION

This thesis has two main topics in the realm of E-shape. The use of E-shape in email analysis and in behavior detection based on accelerometer readings. In relating to email I propose a new method of looking at email. This method both takes content and context out of the classifier and uses how the email "looks" to a human eye. I then propose a method to use the shape to classify botnet into different groups. I showed an initial accuracy of 80%. I then continued the effort looking at spam and ham classification and how well the E-shape model performed, and finally looking at an individual email writing style.

The second part of the thesis looks at applying the foundations of E-shape analysis to look at something all together different: accelerometer readings. In this thesis I will show the accelerometer reading can be applied to a persons daily activities, such as walking or driving, and the look at the application of E-shape to quantify a day.

1.1. E-Shape Concept

The concept of E-shape, I believe, is much broader than what is included in this thesis. I think that everyone's activities will have a shape in either or both the temporal and spatial domains. For example: writing emails, driving, texting, phone calls, walking or any combination. The shapes produced could be use to quantify the person and provide a parameter to be used in classification.

1.2. Previous Work Related to Botnet Detection

Botnet detection has been for many years now. Each paper on the subject seems to be an innovative new technique. Botnet pose a significant security risk because of their size and difficulty to track based on its distributed nature. From DNS black hole detection to using honeypots to track them. The work that lead to E-shape came from detection based on header analysis. Header analysis is used to track where emails are coming from, where they are going and where they have been. Information contained in headers is very useful when hand labeling emails for botnet detection. During my progress expanding the work of header analysis, which included hand labeling of thousand of emails into botnet buckets, came the idea of E-shape. I found several patterns based on how emails looked which could directly be applied to which botnet the email belonged to.

1.3. Applications of E-Shape

Email spam is still the largest contributor to internet mail that is being sent. Most of that email is sent from entities called botnets. The first section of this thesis lays out an innovative approach to dealing with botnets and spammers. Spam filters of today are very powerful, but when a spammer comes up with a new approach to beat a filter, its spam shape can still be classified.

In my look at E-shape analysis I stated earlier that there were three main contributions of this work: botnet detection, spam and ham labeling, and individual email finger print analysis. My look at botnet detection goes through the process of hand labeling a corpus of emails. Next, designing an algorithm to handle the shapes and classify them together. Finally, I show a table of results including false positive and false negative rates.

After such a successful run at botnet detection I decided to turn the powerful concept of E-shape to that of spam and ham detection. The motivation is that most classifiers rely on training of known spam messages. As happens often, spammers are able to get several rounds of spam through classifiers before they are noticed. I propose using E-shape analysis in a complimentary fashion to current classifiers. The idea is that if classifiers had a no content or context analysis of an incoming email it might have a better chance of classifying unknown messages more accurately. My technique shows promising results at near 70% success with spam and ham labeling.

Lastly, if E-shape was able to classify botnets with such high accuracy then what do people look like on this scale? In this section I have done preliminary work on the analysis and classification of the top three senders in a single user's inbox. In this work, I am able to show an accuracy of 75% by applying the same principles from botnet detection.

1.4. Main Contributions

The main contribution of this thesis is the concept of E-shape. In this thesis I describe several uses for E-shape: email botnet analysis, email spam analysis, email individual fingerprint analysis and behavior analysis based on accelerometer readings. I show how to use E-shape as a tool in email and how to classify patterns in daily activity generation.

1.5. G1 Experiments

The G1 provides a means to measure several more sensors than a standard phone, and even one-ups the iPhone with an internal compass. Several applications are able to take control of these sensors to do amazing things, such as point the phone at the stars to see a constellation map of the area the phone is pointing at. I am going to use these sensors for activity detection such as walking or working at the desk. Then applying the data over a long period I will provide an E-shape analysis in hopes to classify or distinguish individuals specifically. For example, Subject A works out in the morning, goes to class, walks (a lot), and then goes to sleep, whereas Subject B will wake up late, drive around, work at his desk at school, work at his desk at home, and then sleep. I then apply the technique of E-shape to show the similarities in behavior from one day to the next, including Monday to Monday and so on.

CHAPTER 2

E-SHAPE ANALYSIS OF EMAIL

2.1. Introduction

The behavior of email is something that is often overlooked. Email has been with us for so long that I begin to take it for granted. However, email may yet provide new techniques for classification systems. In this paper, I introduce the concept of email shape analysis and a few of its applications. Email shape analysis is a simple yet powerful method of classifying emails without the use of conventional email analysis techniques which rely on header information, hyperlink analysis, and natural language processing. It is a method of breaking down emails into a parameterized form for use with modeling techniques. In parameterized form the email is seen as a skeleton of its text and HTML body. The skeleton is used to draw a contouring shape, which is used for email shape analysis.

One of the largest threats facing the internet privacy and security of email users is spam email. According to the *NY Times* in March 2009, 94% of all email is spam. Email can contain malicious code and lewd content, both of which need to be avoided by 100%. The use of a behavior based detection method will increase the accuracy and compliment current analysis methods in malicious and spam activity.

In this paper, I discuss a case study involving spam botnet detection. I also discuss the possible applications spam and ham filtering and social finger printing of senders. Recent papers presenting on this topic of botnet detection use network traffic behavior [11, 18] and also domain name service blackhole listings [12], whereby botnets are discovered when they query the blackhole listings in domain name system (DNS) servers. By introducing shape analysis, one can further confirm the authenticity of the bot classifier.

The first application goes back to the proverbial spam question [15, 5, 6, 21]. I look at the ability of shape analysis to correctly identify spam. In this study I are not trying to compete against the Bayesian filter, but rather compliment its decision process by offering non-content and non-context aware classification. The nature of the shape analysis classifier allows for both language independent and size independent email shape generation. This is believed to be very useful as the world becomes further integrated and spam comes in multiple languages to everyone.

In the second application, I look at the potential of email shape analysis to identify social context-based finger prints. I propose the ability to distinguish individual or group senders based on the social context. The data set for this study is one subject's personal email inbox.

The rest of the paper is organized as follows. The concept of the proposed email shape is described in Section 2. Section 3 presents the case study email spam botnet detection. Section 4 discusses future work and their preliminary results on email spam filtering and social context-based finger print identification. Section 5 reviews some limitations of my study. Section 6 concludes the paper with a summary and an outlook on future work.

2.2. Email Shape

I define "shape" of an email as a shape that a human would perceive (*e.g.*, shape of a bottle). Without reading the content, shape of an email can be visualized as its contour envelope.

Email shape (E-shape) can be obtained from its "skeleton" that is simply a set of character counts for each line in the text and HTML code of email content. Let L denote the total number of lines in the email text and HTML code, and h_k denote the character count (this includes all characters and whitespace) in line k. A skeleton (H) of an email thus can be defined as follows.

$$H = \{h_1, h_2, h_3, ..., h_L\} . \tag{1}$$

Skeleton H can be treated as a random variable. Thereby the shape of an email can be derived from its skeleton by applying a Gaussian kernel density function (also known as Parzen window method) [9], which is a non-parametric approach for estimating probability density function (pdf) of a random variable and given by Eq. 2.

$$f(x) = \frac{1}{Lw}\sum_{k=1}^{L} K(\frac{x-h_k}{w}), \tag{2}$$

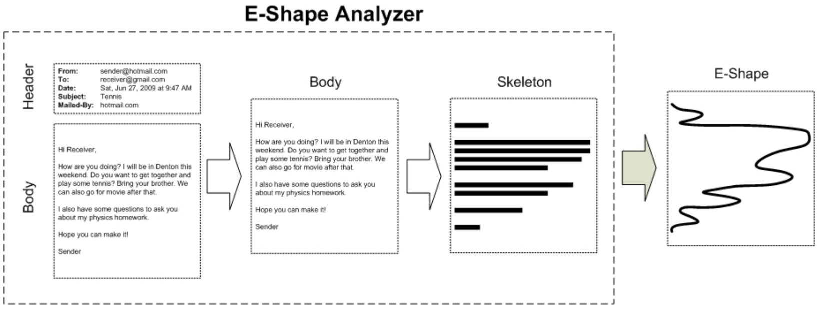
where $K(u)$ is the kernel function and w is the bandwidth or smoothing parameter. To select the optimal bandwidth, I use the AMISE optimal bandwidth selection based on Sheather Jones Solve-the-equation plug-in method [14]. My kernel function is a widely used zero mean and unit variance given by Eq. 3

$$K(u) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2}. \tag{3}$$

With this approach, an algorithm for finding E-shape can be constructed as shown in Alg. 1. Figure 2.1 illustrates the process of extracting E-shape. An example of four different E-shapes is illustrated in Fig. 2.2.

In summary, email shape is found by computing the number of character per line in an email. Almost every email has a text and HTML body. The lines are put into a file from which the Gaussian kernel density estimator smooths the rigid line graph into a normalized, smoothed graph. This graph is calculated for every email. I then performed a comparative function, called Hellinger distance, to find how closely each email shape is related.

Figure 2.1 Email shape analyzer.



From: sender@hotmail.com
To: receiver@gmail.com
Date: Sat, Jun 27, 2009 at 9:47 AM
Subject: Tennis
Mailed-By: hotmail.com

Hi Receiver,

How are you doing? I will be in Denton this weekend. Do you want to get together and play some tennis? Bring your brother. We can also go for movie after that.

I also have some questions to ask you about my physics homework.

Hope you can make it!

Sender

Hi Receiver,

How are you doing? I will be in Denton this weekend. Do you want to get together and play some tennis? Bring your brother. We can also go for movie after that.

I also have some questions to ask you about my physics homework.

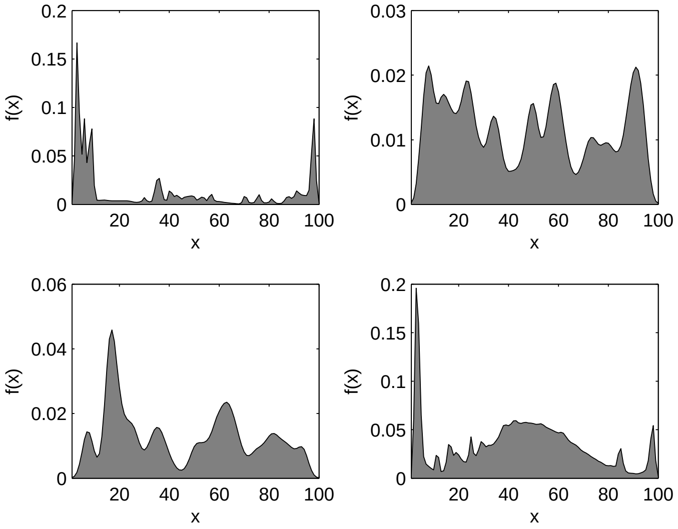
Hope you can make it!

Sender

Algorithm 1. Email Shape

```
$S$ = Email Shape($C$)
Input: Email Text and HTML code ($C$)
Output: E-shape ($S$)
FOR $i = 1$ to $L$ /*$L$ is the total number of lines in email HTML code */
    $h_i$ = character count of line $i$;
END FOR
$H = \{h_1, h_2, h_3, ..., h_L\}$; /* skeleton is extracted */
$S$ =applying Gaussian kernel density function on$H$; /* E-shape is obtained */
Return $S$
```

Figure 2.2 An example of four different E-shapes.



2.3. Applications of E-Shape

My understanding of what shape analysis has to offer to the community is only at the beginning. I present, in the paper, a case study and two future work applications that outline some of the behaviors that shape can be used to analyze. First, is analysis of spamming botnets by template and/or campaign detection based on shape. By identifying similar shapes from different parts of the globe, one could surmise that they come from a matching bot host controller. (A bot is a compromised host that resides on the internet, usually without the host's controller's knowledge. The term bot has negative connotation and is usually associated with malicious behavior such as spamming, denial of service, or phishing. A botnet is a collection of two or more bots, and sometimes on the order of 10,000.) Second, spam filtering has become second nature to world. It has over 99% accuracy, but what of the last less than 1%? What were the content and context that were able to escape the filtering process? In this application I propose that E-shape analysis can be used to get closer to the goal of 100% spam classification. Third, E-shape analysis shows the discriminatory power to identify individuals on a personal level. In this application I build personal finger prints and turn my classifier over to the ham side of email.

2.3.1. Spam Botnet Detection

Spamming botnets are notoriously hard to pin point, often needing to use several methods to achieve decent accuracy. Here I present another tool to use in the assessment of botnet detection. For this case study I gathered a data set of spam emails collected by Gmail's spam filter over the period of one month, during July 2008. The data set was over 1,100 emails in four different languages. The majority language was English. This data set was hand labeled into buckets based on content, size, and email type (e.g. Plain, HTML, Multipart). Each bucket would then contain similar emails, for example one group would contain emails sent that contained "Kings Watchmaker".

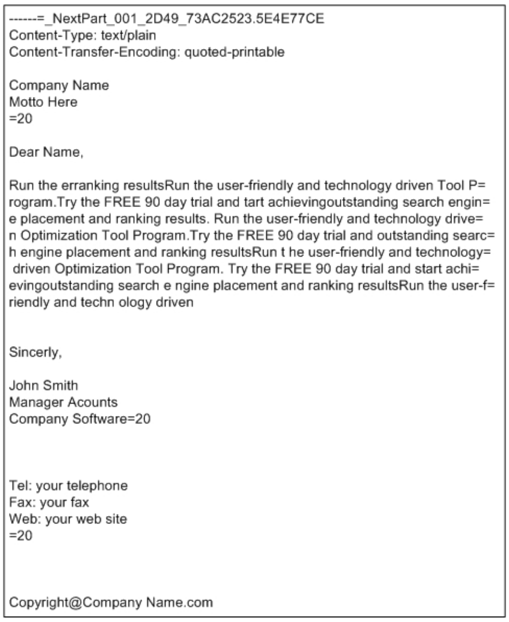
2.3.1.1. *Hand labeling.* To hand label thousands of emails I developed a program to display emails for ease of labeling. The program allows for a user to view a recorded history of previous labels, at any time refer to specific email for comparison, and resume previous labeling sessions. Files are written to an object text file, known as pickling, to preserve the email object format. The botnet label is written as a header directly into the email. A graphical user interface is included for the program.

After labeling several hundred of the emails, I started to see patterns emerge. I found evidence to support that botnet spammer's used templates to bypass spam filters, and they would fill in the blanks with the links and info they needed to get through (An example of the actual spam botnet template is shown in Fig. 2.3). The spam emails are very diverse, also shown by the multiple languages. The details of my data set is listed on Table 2.3.1.2.

2.3.1.2. *Template discussion.* In the United States over 650 million email accounts are owned by four companies: Microsoft (MSN), Yahoo, Google and AOL [3]. Google comes in a distant third to MSN and Yahoo. They are very protective of their users and to get solicited emails to them can be an expensive process. I have evidence [10, 16, 19] to believe botnets are using specific templates to beat out spam filters. Seen in Fig. 2.3, a spammer would simply need to fill in the blanks and begin his campaign. The use of randomized or individually written emails for the purpose of spamming is not feasible on any small, medium or large scale campaign. It is of note to the authors that multiple botnets could be using the same template and be classified together. A separate method will be analyzed for distinguishing them in future work.

The total number of buckets from hand labeling was 52. For analysis I discarded buckets that had less than 10-emails per. This yielded 11 buckets. The shape of the testing email was derived using Alg. 1, then classified into different botnet groups. The measure of difference in shapes between these groups was based on Hellinger distance [4] since the E-shape is built with an estimated probability density functions (pdf). By using an estimated pdf, I am able to smooth out the shape from its rigid skeleton. It also normalizes the number of lines in the email, for use of Hellinger distance. The normalization of length is what provides a size independent way to calculate shape. Looking at the template in Fig. 2.3, a host spammer could add another paragraph with more links and not still not drastically change the normalized E-shape of itself.

Figure 2.3 An example of the actual spam botnet template.



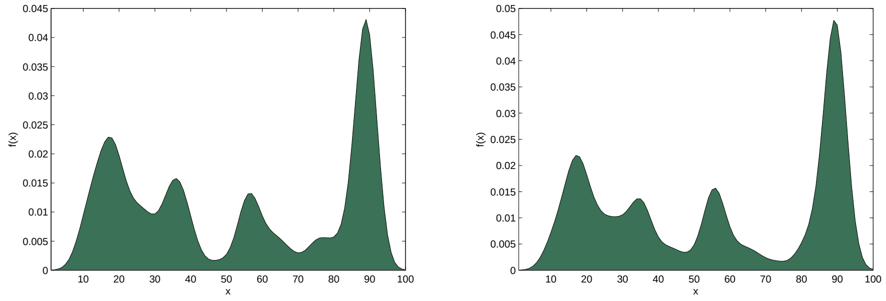
```
------=_NextPart_001_2D49_73AC2523.5E4E77CE
Content-Type: text/plain
Content-Transfer-Encoding: quoted-printable

Company Name
Motto Here
=20

Dear Name,

Run the erranking resultsRun the user-friendly and technology driven Tool P=
rogram.Try the FREE 90 day trial and tart achievingoutstanding search engin=
e placement and ranking results. Run the user-friendly and technology drive=
n Optimization Tool Program.Try the FREE 90 day trial and outstanding searc=
h engine placement and ranking resultsRun t he user-friendly and technology=
 driven Optimization Tool Program. Try the FREE 90 day trial and start achi=
evingoutstanding search e ngine placement and ranking resultsRun the user-f=
riendly and techn ology driven


Sincerly,

John Smith
Manager Acounts
Company Software=20



Tel: your telephone
Fax: your fax
Web: your web site
=20



Copyright@Company Name.com
```

Table 2.1. Details of dataset for botnet detection experiment.

Feature	Count
Total Emails	1,144
Email's Sizes of 1 to 100 lines	906
Email's Sizes of 101 to 200 lines	131
Email's Sizes of 201 to 300 lines	42
Email's Sizes of 301 to 400 lines	25
Email's Sizes of 401 to 500 lines	40
Emails in English	815
Emails in Chinese	270
Emails in Spanish	57
Emails in German	2

Figure 2.4 shows two email shapes from a Chinese botnet. Figure 4(a) is larger than Fig. 4(b) by 22 lines, a difference of 11.8%. The two shapes are considerably similar and were mapped to the same bucket by the E-shape algorithm.

Figure 2.4 Showing size independence of shapes from the same botnet.



(a) Shape 1, 186 lines

(b) Shape 2, 164 lines

The signature of each botnet group was computed as the expected value (mean) of the group. I used predefined threshold level at 0.08, which found to be the optimal threshold for my study. Hellinger distance is widely used for estimating a distance (difference) between two probability measures (*e.g.*, pdf, pmf). Hellinger distance between two probability measures A and B can be computed as follows.

$$d_H^2(A, B) = \frac{1}{2}\sum_{m=1}^{M}(\sqrt{a_m} - \sqrt{b_m})^2, \tag{4}$$

where A and B are M-tuple $\{a_1, a_2, a_3, ..., a_M\}$ and $\{b_1, b_2, b_3, ..., b_M\}$ respectively, and satisfy $a_m \geq 0, \sum_m a_m = 1, b_m \geq 0$, and $\sum_m b_m = 1$. Hellinger distance of 0 implies that $A = B$ whereas disjoint A and B yields the maximum distance of 1.

The accuracy of this data set is found from computing the number of correctly labeled emails in a bucket to the total number of emails in that bucket. A false positive indicates an email that was placed in the bucket but did not belong. A false negative would be the total number of emails, from hand labeling, that are in the rest of the buckets which belong to that bucket.

Figure 2.5 A result of the botnet detection experiment based on 879 different size and language emails.

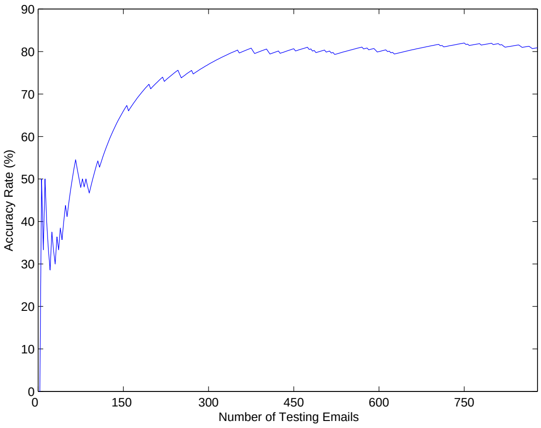


Figure 2.5 shows a promising accumulative accuracy rate of almost 81%. This number reflects the cumulative accuracy of all the buckets. While some buckets have a low accuracy, several of the buckets have a very good accuracy up to and including 100%, seen in Table 2.2. The evidence of a 100% accuracy bucket would show a positive match on an email campaign template. Accuracies below 50% are simply emails that are of similar shape. For example, bucket 6 is a mismatch of several botnet's from different languages and types of spam emails. Email shape analysis is showing good results in botnet and campaign classification, the purpose being to take context and specific content out of the classification process.

2.4. Future Work and Preliminary Results

In my on going work to discover and explore the full potential of E-shape analysis, I take a look at a couple of possible applications and also some preliminary analysis and results on them. Below I discuss the use of E-shape on spam filtering and on social context-based finger printing. In my finger print analysis I look at the capability of E-shape to differentiate senders from each other.

Table 2.2. Accuracy of individual bucket.

Bucket	Accuracy	False Negative	False Positive	Total Emails
1	41.37%	14	48	81
2	74.07%	20	20	76
3	80.95%	20	11	59
4	100%	0	0	129
5	68.75%	0	28	90
6	45.83%	0	25	67
7	100%	0	0	78
8	93.10%	14	6	81
9	88.00%	22	17	140
10	100%	0	0	118
11	100%	0	0	70

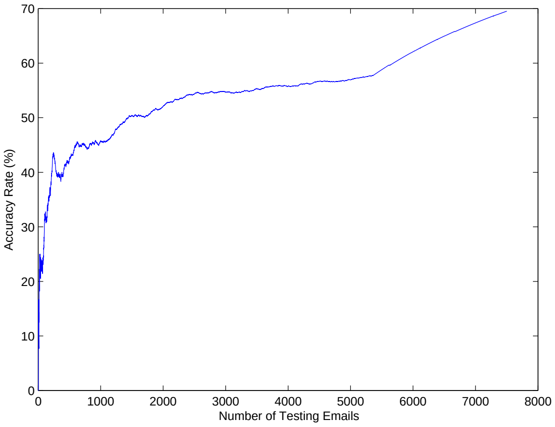
2.4.1. Spam Filtering

In this application of E-shape, I discuss the behavior that E-shape analysis can have on the spam filtering process. The Bayesian filter proves to be over 99% successful most all the time. However, to reach the goal of 100% further analysis is required. The Bayesian filter uses content and context to classify emails. The process could be enhanced using the method of shape analysis to “look at” if an email is spam or ham, taking content and context completely out of the equation. Surprise emails to the classifier that can’t be categorized or are unique in manufacturing might make it through.

The data set used for this case study was the Trec 2007 corpus [7]. The Trec corpora are widely used in spam testing. The 2007 corpus was over 74,000 emails. However, for this study, only the first 7,500 emails were used for analysis. The corpus was approximately 67% spam and 33% ham and has been hand labeled by the Trec Team.

The method for comparison of spam versus ham was similar to that of botnet detection case study. Here I again used an unsupervised learning algorithm to classify data. I have developed a program that will take an email file in MBOX format and calculate how many similar groups their are and classify in the same way as section 3.1. A testing email was classified to ham or spam based on the closest clustered group signature. The drawback of this process is that the buckets will need to be labeled by ham or spam, which is independent of my classifier. Once the bucket is known to be ham/spam future email's which are classified into the bucket will be labeled as such.

Figure 2.6 A result of the spam filtering experiment based on 7,500 emails from TREC 2007 corpus.



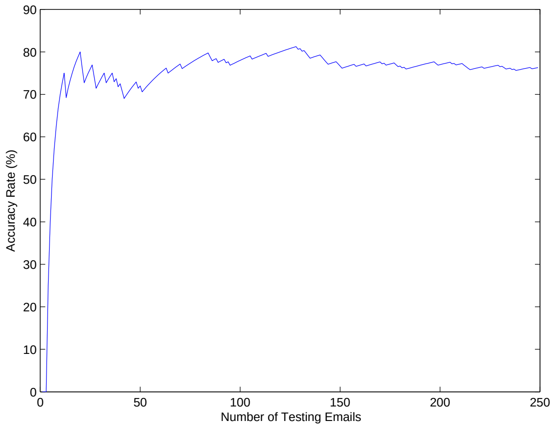
Preliminary results show an accumulative accuracy of about 70% for 7,500 emails. The accuracy is great considering that no content or context was even referenced. The ability for shape analysis to act as a spam filter would be recommended for use with emails that the Bayesian filter finds unsure about. Future work in that regard would be to implement shape analysis inside the Bayesian filter process.

2.4.2. Social Context-Based Finger Print Detection

This application is on using E-shape analysis to identify an individual's personal email finger print based on social context. I define personal fingerprint as the shape that one typically uses to contact others with. When an individual writes emails, it is believed that his shape will stay relatively the same, although length may change, the way he/she writes will not. An example of this would be an individual that creates a new line about every 40-50 characters versus a person that creates no new lines at all. It is also believed that this method can be used to reveal a user's cliques, as seen in [17]. A user will type differently to his/her boss and work mates than he/she would to their close friends. In this case study I follow the aggregate pattern of other users sending to a specific person.

For the data set, I used the top three different senders from one subject's inbox. The emails were collected over five months. Using E-shape analysis, I was able to distinguish these three senders to this subject, from unaltered emails (no thread deletion), with an accuracy of about 75% (see Fig. 2.7). The accuracy is considered good. Further refinement and post processing will be looked into in the future for better results. The current results now is using only the E-shape analysis.

Figure 2.7 A result of the social context-based individual's personal email finger print experiment based on three different individual email senders to a subject with total of about 250 emails.



Of the approximately 250 emails that are tested and of the three groups selected was a bi-weekly newsletter from a sales web site. The emails that came from this web site were classified with 100% accuracy and no false positives. The other two senders were from real human conversations.

This method reveals a very powerful tool in categorizing incoming emails when comparing non-human to human emails. Newsletters, advertisements, and solicitations can be moved separately by themselves to be reviewed later by a user, keeping priority emails displayed first.

2.5. Limitations of the Study

Currently the E-shape analysis tool does not have a way to compliment its decision process by removing email threads and conversations. This drawback is reduced by the power of E-shape analysis, but it believed that I still yet have many abilities to unlock in this regard.

The shape analysis is a very useful tool to complement other tools as it can provide the deciding factor to many close decisions. Such is the example in spam detection where the content classifier can already achieve such a high accuracy. Some emails are short by nature, the ability for shape analysis to distinguish between others becomes limited. In the case of spam, short emails are common and the limitation impact of E-shape analysis will be mitigated to a large extent.

As mentioned earlier, and with any tool, the less information you give it, the less it can tell you. In the study of social context-based finger print detection, if a subject has a subset of friends that like to send web hyperlinks back and forth, the classifier will be unable to distinguish between users. Study of group based social awareness could be a possible application of this research.

Botnet detection is a challenging problem. There is not a singular solution to this threat, and combining the latest innovations only brings us a step closer. The purpose of E-shape analysis for botnets is to bring the world one step closer. E-shape analysis is a tool capable of template/campaign identification to find a spamming bots before they are even able to send. Botnet identification is the next logical step of the process and can be supported with this tool.

2.6. Conclusion

In this paper, I present a novel concept of email shape (E-shape) and discuss three case studies using a hidden discriminatory power of E-shape. By using E-shape analysis I was able to detect botnet template/campaigns with about 81% accuracy. The botnet analysis can also be done with multiple languages and email sizes, which shows that the E-shape analysis is language and size independent. Next, I discuss the capabilities of E-shape in spam filtering. Since E-shape is neither content nor context aware, it provides a unique point of view when looking at spam emails. I used the TREC 2007 corpus to test the spam filtering capabilities of E-shape. After running 7,500 emails through the email shape detector, I had a success rate of about 70%. Lastly, I looked at social context-based finger print detection, where I analyzed a single subject's email inbox. Using three different senders, I was able to achieve an accuracy rate of over 70%.

It is important to note that while the accuracy's of my system are not "high," the system of classification is taking content and context out of the classification process. This provides a very useful tool to complement existing methods and tools that currently handle emails, such as inching the Bayesian filter closer to 100% accuracy or assisting network behavior analyzers in determining botnet relationships.

As I evolve my understanding of what E-shape analysis can offer, I plan to improve the accuracies of the existing work and release more case studies. Currently the shape analysis routine does not have any smart way of handling email conversation threads or HTML code. This is the planned next direction of my work and is believed to offer a significant increase to ham labeling accuracy.

CHAPTER 3

G1 INFRASTRUCTURE AND ACCELEROMETER READINGS

3.1. Introduction

In this chapter I will go over the G1 phone more specifically and thoroughly cover the accelerometer. I collect several data sets for the accelerometer and I will be showing what accelerometer readings look like raw, some techniques for simple classification and tables showing results for several daily activities.

3.2. Previous Work in Activity Detection and Accelerometer Readings

I have reviewed three papers so far under this category, they involve the use of activity detection based on bi-axial accelerometer data. The data provided by the G1 phone is uni-axial and is detailed in my use. Some authors chose to use one sensor placed on the hip, then by plotting the data on a X-Y plane, they use features mean, std deviation, skewness, kurtosis, and eccentricity. [1] They plotted a decision tree based on their results for low computational matching of activities. An important contribution of this work is use of the standard deviation to quickly distinguish between static and dynamic activities. Some of the other Features with higher computational values can be completed avoided with they are in the static category of measurements. Experimental histograms from the single bi-axial sensors for eight activities are as follows:

In "Activity Recognition from User-Annotated Acceleration Data" [2] the authors used five bi-axial accelerometers to detect activities. In the end they would conclude that one would only really need two for most a users daily activities. The device is about the size of a G1 phone and records at 76hz. The five sensors were placed on the arm, wrist, hip, thigh, and ankle. Each accelerometer is a 10 G with a 2% tolerance. The measurement period was two sets of 24 hours constant data tracking. The participants did participate in contrived situations such as an obstacle course, strength testing, lying down, etc The results of this paper were over 80% accuracy of about 20 everyday activities. By using a FFT-based feature computation and a decision tree classifier algorithm. In the third paper "Detection of Static and Dynamic activities using uniaxial accelerometers" [20] the use of an accelerometer is applied from a setting of rehabilitation treatment to objective analysis of daily activities. They used a set of two or three uniaxial accelerometers mounted on different places on the body. The thigh and two directions on the chest. The results found were similar to the first paper of easy distinction between static and dynamic activities using the standard deviation. In this data set, a nearest neighbor search can yield results for static data sets. For dynamic data, they used a morphology model to determine the maximum correlation coefficients of individual cycles (or windows). The paper suggests that at least more than one accelerometer will be needed to much better distinguish the difference between some daily dynamic activities [8].

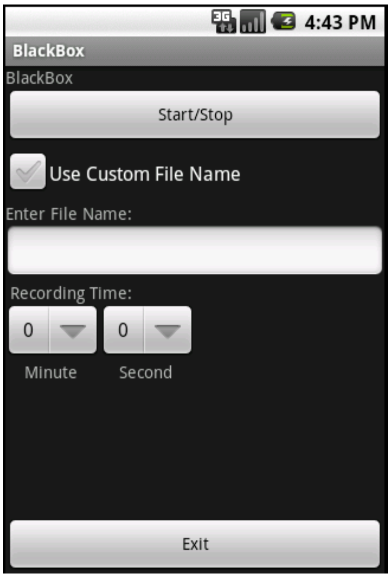
3.3. Problem Definition

The function of this project is to be able to apply G1 sensor data on a broad scale to distinguish individual users based on their activity data, location, and sound. Constraints of the system are battery life of the G1 phone, as applied to continuous data acquisition from the phone, ensuring the phone is on the body at all times, the range and capability of the sensors to perform over long durations, and fudge factors from low quality data acquisitions. This system will be tested on human participants of college age. The testing environment will be the everyday activities of a "generic" student. Including but not limited to driving, eating out, being in class, walking, running, working out, and riding elevators.

3.4. The Android G1 Platform

For the purple of data collection the Android platform was chosen for it's open source nature and ease of programming. A previous student had developed a simple recording

Figure 3.1 Screenshot of blackbox GUI.



☑ Use Custom File Name

Enter File Name:

Recording Time:

0 Minute 0 Second

application for various sensors that are on the T-mobile G1 phone. The sensors include: accelerometer, microphone, GPS, and compass orientation. The application as titled Blackbox. This application was modified for the purpose of this project to collect just accelerometer readings. I added a simple GUI, seen in Figure 3.1, to handle inputs and also added functionality that allowed it to record data periodically throughout a day.

3.5. Experimental Data

The data collected for this experiment is done by dumping the G1 accelerometer data into a file. There are two categories of experimental data. The two categories are: long duration data and short duration specific data.

- Long Duration Data

  Data collected on this duration is set to 25min long. During the experiment I carried out different daily activities such as working at my desk and going to the bathroom, working at my desk and visiting a meeting, walking up and down stairs, riding the elevator. The data collected was over 10,000 points for each accelerometer direction.

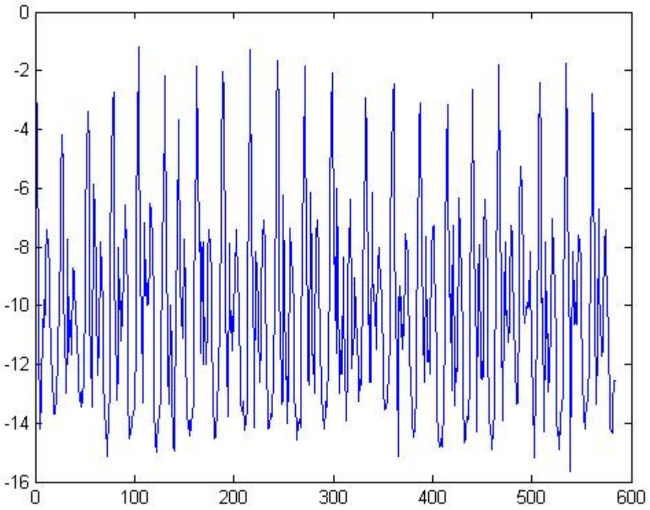
- Short Duration Data

> Data collected on this duration is set from thirty seconds to two minutes currently. The purpose of this data is to use a sliding window technique or correlation signal to identify the difference parts of the long duration data collection. The purpose is to build a chart to identify the separate activities throughout a day. An example being wakeup -> drive to office -> work at desk -> meeting -> lunch .. end day.

The data collection technique provided by the G1 android platform collects data whenever the processor is free. The current variable for the accelerometer sample frequency is set to one. This yields approximately four to five data samples per second.

3.5.1. Walking

Figure 3.2 Walking signal for approximately forty-five seconds.

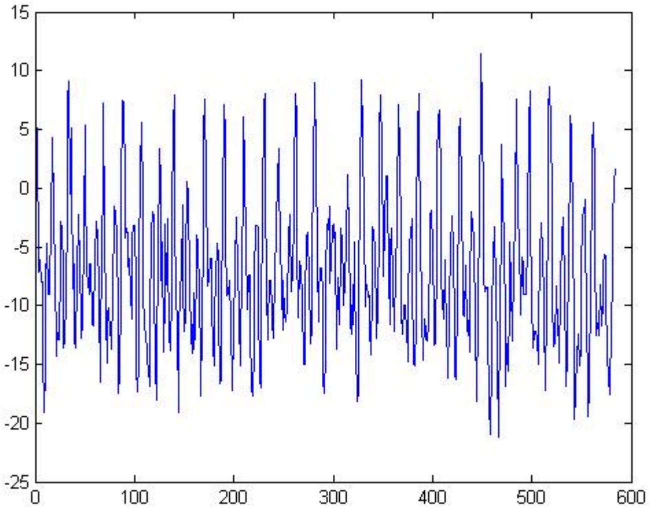


The difference between walking and running is comparable on a signal level. However, from a data level the difference of amplitude between walking and running is significant enough to classify. The mean lower and upper amplitude for walking are -13 to -3. Whereas, in Figure 3.3, the mean amplitude for running can be measured at -15 to 5.

3.5.2. Running

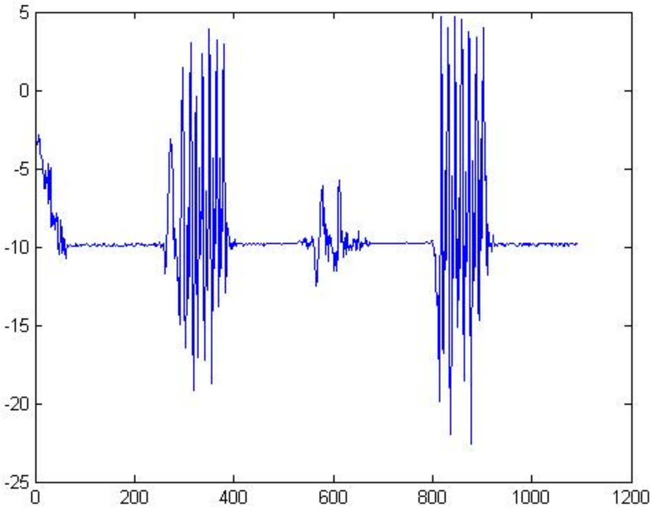
Here is the running data.

Figure 3.3 Running signal (Y-axis) for approximately forty-five seconds.



3.5.3. Upstairs

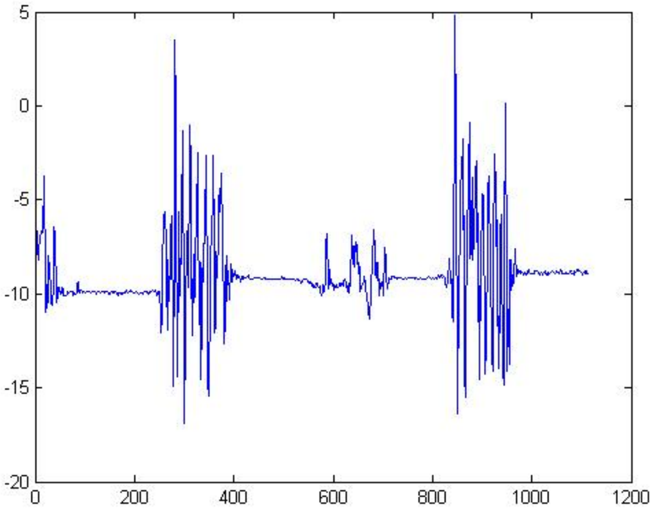
Figure 3.4 Discovery Park stairwell, walking upstairs.



At Discovery Park the stairs are broken into two flights of equal distance separated by a small walking area to the other flight. In the data samples for Figure 3.4 and Figure 3.5, the participant traversed the first set of stairs, paused, then walking to the second, paused and then traversed the second set of stairs. This was done at a normal pace without skipping any step.

3.5.4. Downstairs

Figure 3.5 Discovery Park stairwell, walking downstairs.



3.6. Experiments for Activity Detection

There were three sets of experiments that were performed on the raw data provided by the G1 Andriod platform. They were as follows:

- Signal correlation
- Shape analysis for signal classification
- Signal covariance

The purpose of these experiments is to identify or differentiate daily activities. These activities are:

- Walking
- Running
- Upstairs

- Downstairs
- Standing
- Sitting

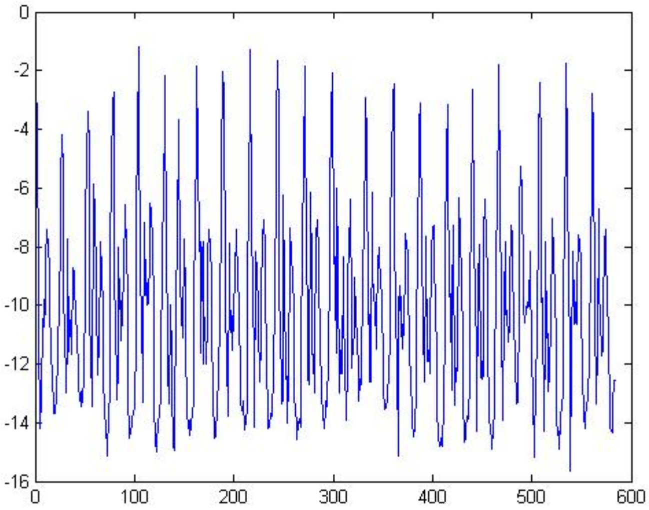
3.6.1. Experiment 1: Signal Correlation

Using signal cross-correlation is a known way to identify how similar two different signals are. It is provided by equation 1 below.

$$(f \star g)(t) = \int_{-\infty}^{\infty} f^*(\tau)g(t+\tau)d\tau \tag{5}$$

In this section, I present a series of graphs depicting several different cross correlation results. However, it is import to know what an exact match would look like for the signals that are used in this paper. For that purpose I have computed and graphed the auto-correlation of the first walking data sample set. It can be seen below in Figure 3.6.

Figure 3.6 Auto cross-correlation of the walking signal.



The most notable feature of the auto cross-correlation is the constant slope until peak and constant slope until zero. With that said, let's move on to the correlation experiment.

In this experiment, as with the other two, I identify the activity based on its cross-correlation to the known data sample for which I want to compare it. For the purpose of experimentation the length of the data sampled are broken into 5, 10, and 25 second segments.

Example high cross-correlations for walking and running are provided below for walking and running. Figures 3.7 and 3.8 show that indeed the signals for comparison to a known data sample space can provide good results. Table 3.1 shows the experiment for comparing various sample sets to each other.

Figure 3.7 Correlation of 15sec of walking to total walking sample.

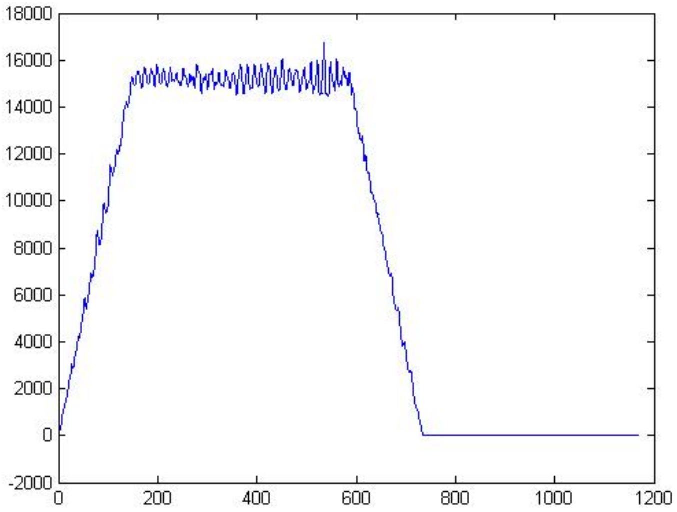


Figure 3.8 Correlation of 15sec of running to total running sample.

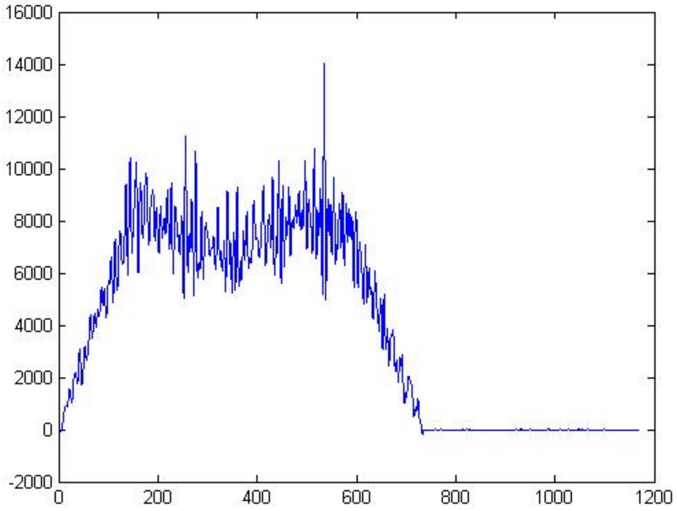


Table 3.1. Cross-correlation of a walking sample validated against the other samples.

Walking 50	walking	running	upstairs	downstairs
y	**2706.806**	1717.569	2235.232	2243.697
z	99.93684	79.92013	117.2196	126.4196
navg yz	42.25485	32.95936	43.43869	32.42289
mag yz	**2973.28**	2527.891	2783.259	2783.004
add yz	**2529.707**	1595.152	1314.905	1001.021
avg yz	**632.4268**	398.7879	328.7262	250.2552
mag xyz	**3028.891**	2626.892	2894.148	2824.842
add xyz	**2161.684**	829.5198	484.9688	819.2707
avg xyz	**240.1871**	92.16887	53.88542	91.03008

Table 3.2. Cross-correlation of a upstairs sample validated against the other samples.

Upstairs 50	walking	running	upstairs	downstairs
y	2119.947	1319.734	1806.653	1757.244
z	98.86776	131.8692	208.879	283.0683
navg yz	51.04611	112.1073	147.2396	166.8779
mag yz	2861.679	2433.008	2678.79	2678.545
add yz	1533.56	978.0301	833.4955	620.219
avg yz	383.39	244.5075	208.3739	155.0547
mag xyz	2947.76	2556.529	2816.625	2749.177
add xyz	763.0843	342.5244	284.3047	346.4859
avg xyz	84.78714	38.05826	31.58941	38.49844

In Table 3.1 and Table 3.2 the values for each is its mean correlation over the entire length of the sample size. The higher the mean, the more and longer I matched the original signal. For this experiment I see that the first running sample set is compared with a walking sample set and the second running sample set. The table shows that although the signals do correlate with each other, I see that the running sample set correlated higher with the walking sample set than it did with the other running sample set. For the purposes of this project, it will not be feasible to use signal correlation on the raw data sample sets for activity detection.

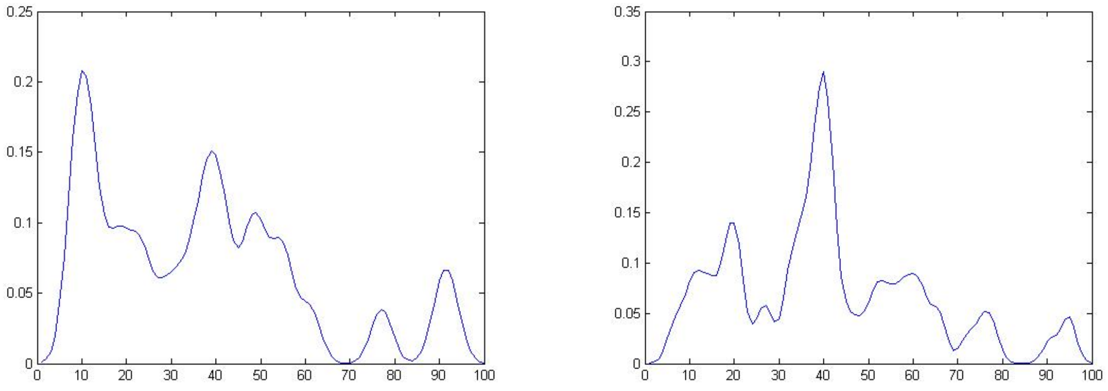
3.6.2. Experiment 2: Shape Analysis for Activity Detection

In my previous work with email analysis, I used a method to determine what an email looked like. This method employed a kernel density function, maximized at a specific bandwidth, and the use of a Hellinger distance to determine how close the functions were related.

$$\widehat{f}_h(x) = \frac{1}{Nh}\sum_{i=1}^{N} K(\frac{x-x_i}{h}) \tag{6}$$

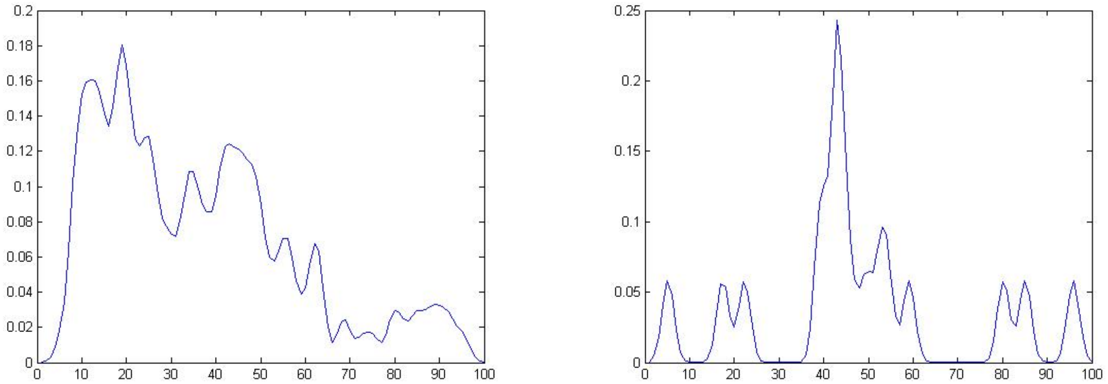
$$H^2(P,Q) = \frac{1}{2}\int\left(\sqrt{\frac{dP}{d\lambda}} - \sqrt{\frac{dQ}{d\lambda}}\right)^2 d\lambda \tag{7}$$

Sample data sets for Shape analysis are done one walking and running data samples only, for proof of concept.



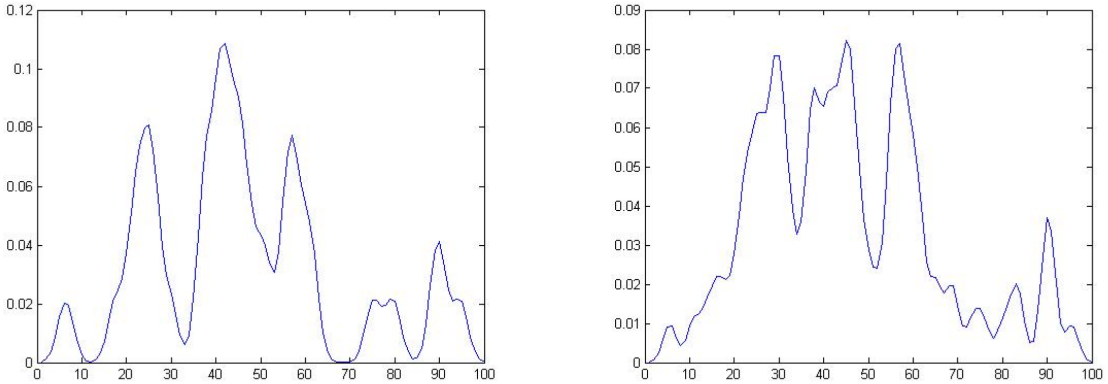
(a) Walking 2 Steps - 5ft

(b) Walking 4 Steps - 9.5ft



(c) Walking 10 Steps - 25ft

(d) Running 1 Steps - 5ft



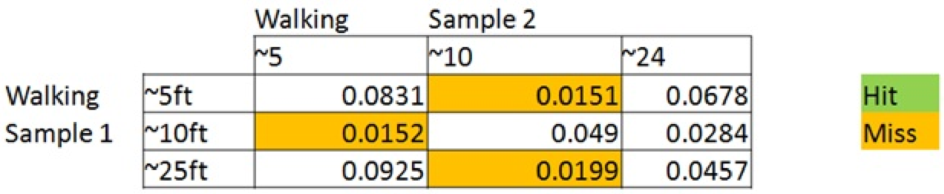
(e) Running 2 Steps - 10ft

(f) Running 5 Steps - 25ft

Figure 9(c) through Figure 9(f) are the data samples over the given length of one to ten steps. They each represent the graph of the signal, over the given duration, run through the kernel density estimator. The data sample lengths are normalized to 100 as a result of the estimation. In tables two through four are the results of this experiment.

In Hellinger analysis, the closeness of two probability functions is found by the output number of the function. This number is between zero and one, where one is not close and zero is perfect match.

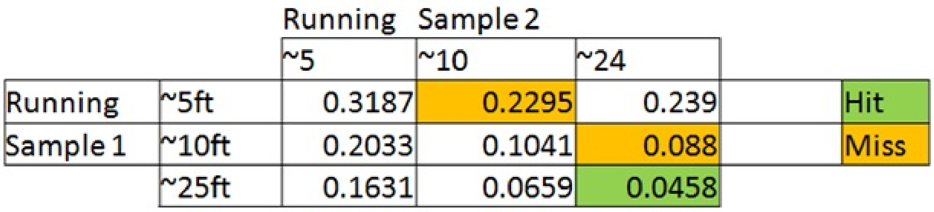
Figure 3.9 Walking Data 1, Hellinger analysis versus Walking Data 2.



		Walking	Sample 2		
		~5	~10	~24	
Walking	~5ft	0.0831	0.0151	0.0678	Hit
Sample 1	~10ft	0.0152	0.049	0.0284	Miss
	~25ft	0.0925	0.0199	0.0457	

In Figure 3.9, one can see that the Hellinger analysis did not 'hit' any of the sample sizes for the purpose of activity detection. If the walking samples can't even match themselves they can't be used for the purpose of matching other activities.

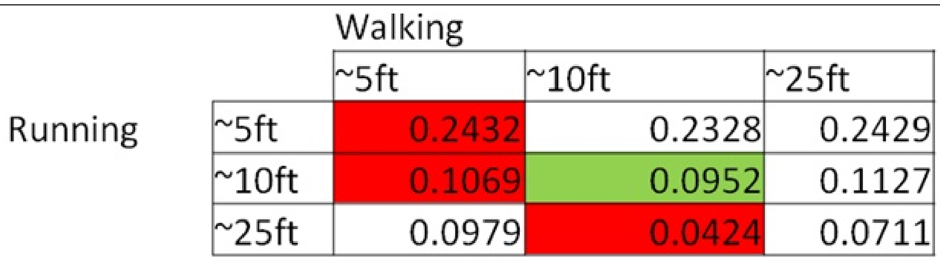
Figure 3.10 Running data 1, Hellinger analysis versus Running Data 2.



		Running	Sample 2		
		~5	~10	~24	
Running	~5ft	0.3187	0.2295	0.239	Hit
Sample 1	~10ft	0.2033	0.1041	0.088	Miss
	~25ft	0.1631	0.0659	0.0458	

In the running sample set I did match on the highest closeness value for that of 25ft or ten steps. However, it is only one match out of three and its match is followed closely behind the 25ft vs 10ft of the same row.

Figure 3.11 Running sample versus walking sample; Hellinger analysis.



		Walking		
		~5ft	~10ft	~25ft
Running	~5ft	0.2432	0.2328	0.2429
	~10ft	0.1069	0.0952	0.1127
	~25ft	0.0979	0.0424	0.0711

In Figure 3.11 for Hellinger analysis of shape functions, one can see that walking and running do not provide any promising results. The technique did match the 10ft category, but it's more likely that it is a false positive as it too close to another category to be used for comparison.

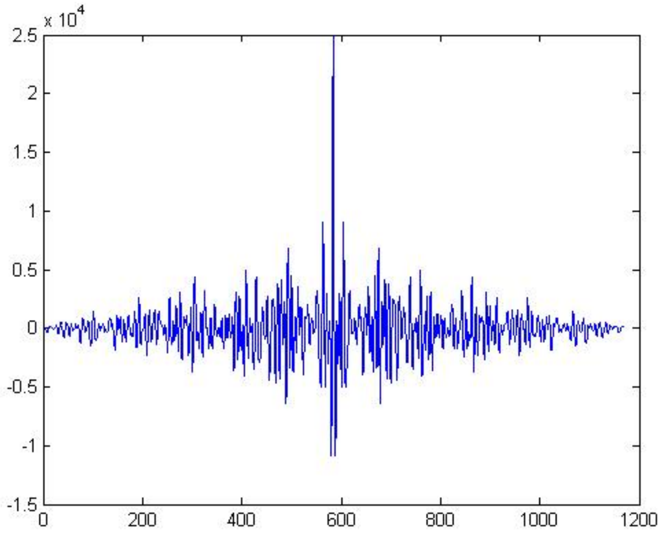
Shape analysis for the purpose of activity detection is not valid.

3.6.3. Experiment 3: Signal Covariance

As with signal correlation, signal covariance (or cross-covariance) is a method to determine signal similarity. However, what makes it different is that it finds the instantaneous similarity for every sliding position. For the purpose of this project, it has been found that it will yield more information, as I compare smaller data sample sets with larger ones. The equation for cross-covariance is below:

Let us take a look at the walking sample set auto cross-covariance, in Figure 3.12. One could see as the whole sample set slides past itself I have some smaller matches as I approach the middle and perfect match peak, the other half is simply a mirror when comparing data with itself.

Figure 3.12 Auto cross-covariance of 45sec walking data set.



In this experiment, I use the mean of the cross-covariance, however, with the addition of the excluding everything below zero from the average. That way only high similarity values are compared. This method is used because, for the experiment, I do not care how much the signals do not much, but rather only how much they do. In Figure 3.13, one can see that fifteen second time duration matches dozens of times to the total walking sample set. This was expected.

Figure 3.13 Cross-covariance walking 15sec to total sample set.

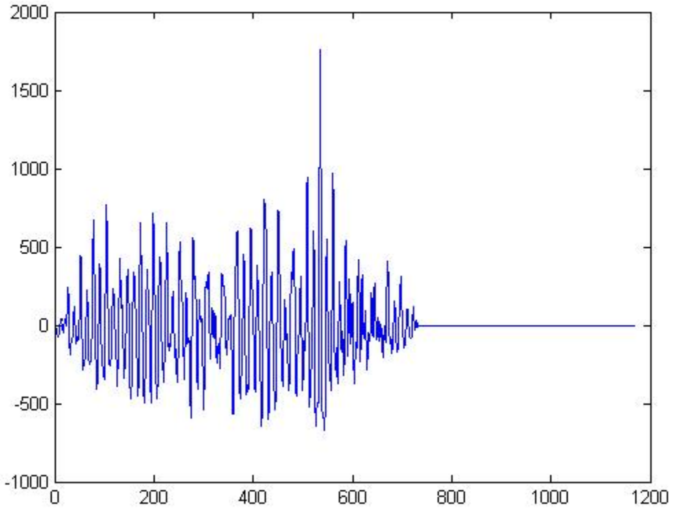
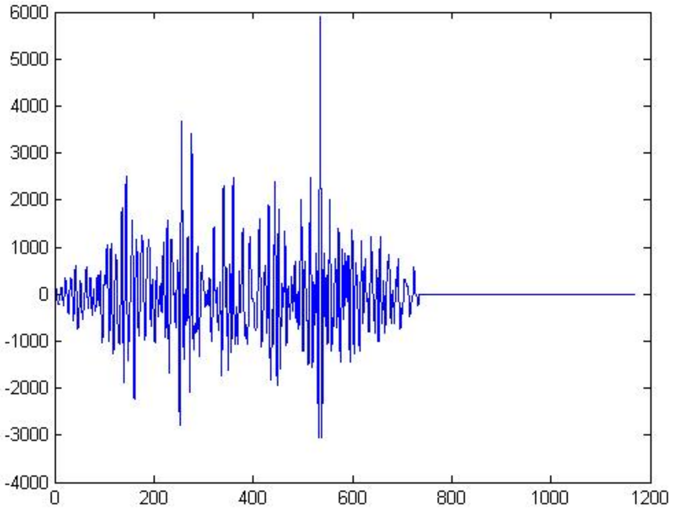


Figure 3.14 Cross-covariance running 15sec to total sample set.



In figure 3.14, the same observation can be made. The ability for cross-covariance to match running with itself seems to valid, but let's take a look at the experimental results to tell for sure. The cross-covariance of the walking sample to the downstairs sample can be seen in Figure 3.15. Going back to Figure 3.14 where running was compared with itself, the peak match was almost 6000. Here, the peak barely reaches 600. This shows a low level of similarity.

Figure 3.15 Cross-covariance walking to downstairs sample sets.

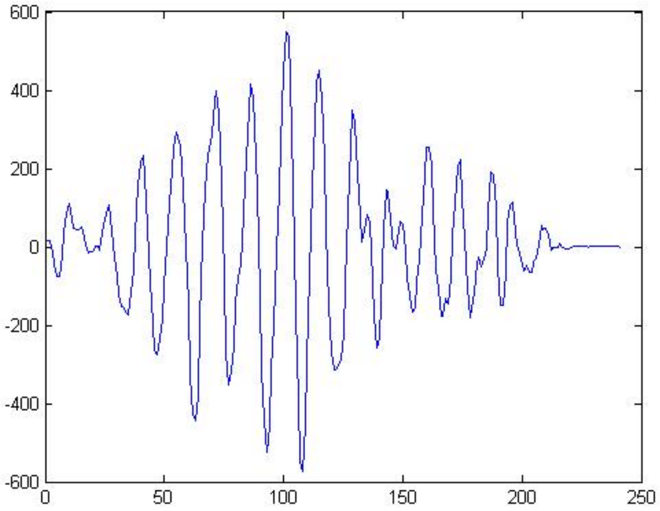
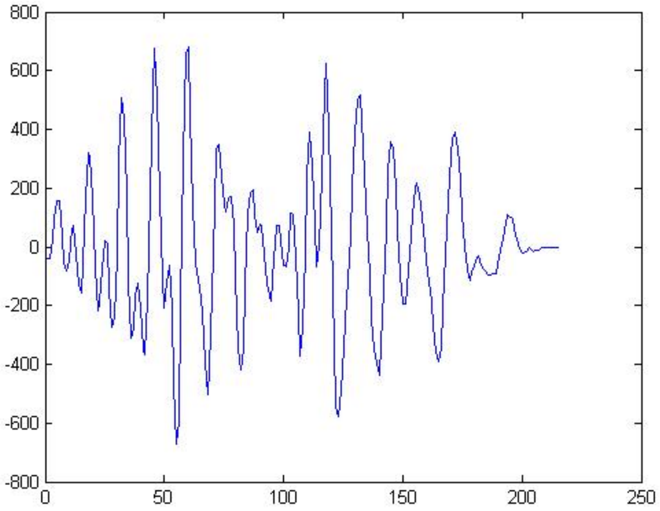


Figure 3.16 Cross-covariance walking to upstairs sample sets.



One can see a similar result for the walking compared to the upstairs, Figure 3.16, traversal. The peak here is now close to 700, but that is still nowhere near the 6000, as seen from highly correlated signals.

3.6.3.1. *Mean Cross-Covariance above 0 results.* The mean cross-covariance proves to be very accurate in the classification of activities from the raw data input.

Table 3.3. Mean Cross-covariance above 0; Running Sample 1.

Running 50	walking	running	upstairs	downstairs
y	126.9562	210.0808	274.5374	150.918
z	103.4553	**123.3506**	117.6539	60.53484
navg yz	37.96293	**52.3042**	51.6638	42.53343
mag yz	65.3918	93.84883	117.3031	62.88274
add yz	151.8517	**209.2168**	206.6552	170.1337
avg yz	37.96293	**52.3042**	51.6638	42.53343
mag xyz	65.40826	94.27359	122.4967	67.46729
add xyz	94.56421	**154.3987**	102.5159	97.07261
avg xyz	10.50713	**17.15541**	11.39066	10.78585

Table 3.3 compares the similarity of running sample set 1 to the walking, running sample 2, upstairs and downstairs sample sets. In this experiment, it had 100% accuracy.

Table 3.4. Mean Cross-covariance above 0; Walking Sample 1.

Walking 50	walking	running	upstairs	downstairs
y	92.06971	92.59904	154.9018	66.50844
z	**99.94654**	79.01264	68.29057	56.05774
navg yz	**40.94989**	31.86386	40.48347	20.76497
mag yz	**99.52707**	71.41895	145.5966	76.87946
add yz	**163.7996**	127.4554	161.9339	83.05988
avg yz	**40.94989**	31.86386	40.48347	20.76497
mag xyz	97.18249	72.11653	141.833	75.48664
add xyz	**264.3535**	217.4726	189.4907	94.93926
avg xyz	**29.37262**	24.16362	21.05452	10.54881

In table 3.4, I look at walking sample 1 versus the other sample sets. In this table one can see that in the 5 second row, I missed the comparison. This suggests that my sample size might need to stay above the 5 second bracket.

Table 3.5. Mean Cross-covariance above 0; Upstairs Sample 1.

Upstairs 50	walking	running	upstairs	downstairs
y	253.9436	224.3018	**367.6658**	176.8175
z	97.95661	143.138	81.20733	76.86471
navg yz	55.5811	40.44023	**58.69169**	45.0502
mag yz	192.9463	126.3151	**248.7215**	118.2105
add yz	222.3244	161.7609	**234.7668**	180.2008
avg yz	55.5811	40.44023	**58.69169**	45.0502
mag xyz	178.4674	128.134	**240.5808**	117.6227
add xyz	243.1239	198.5676	222.2814	171.4306
avg xyz	27.01377	22.06307	24.69793	19.04785

Table 3.5 is a comparison of the upstairs staircase sample sets compared to the all of the sample sets. The upstairs sample set had very promising results; however, the downstairs sample set didn't do as well. This also suggested that a longer sample space might be needed.

3.7. Conclusion

In this paper I showed several methods to process the raw accelerometer readings from the Android G1 Phone. The first of which was Signal Correlation. Correlation did show high similarity for signal processing however it wasn't able to effectively distinguish the difference between the difference activities. Next was kernel density estimation combined with Hellinger distance for finding similarity. This was found to be ineffective as well. The last method was signal cross-covariance. After finding the cross-covariance I took the mean of all the data above 0. This gave us a very useful way to differentiate the daily activities. It was able to find every effectively except for downstairs traversal. I then look at an introduction to shape analysis of the daily activities. The purpose is to model a person and his/her days worth of activities. Future work will be expanding the model to include the ability to find all daily activities and furthering the progress of E-shape analysis on the data sets.

CHAPTER 4

DAILY BEHAVIORAL SHAPE

4.1. Introduction

This paper underlines the usage of accelerometers to log different activities and build E-shapes (described later) to identify a persons daily routine. The purpose is to be able to recognize and distinguish six different activities from a single user. These activities are walking, running, upstairs, downstairs, sitting, and driving. The activities are recorded periodically throughout a day. The recordings then provide a means to build patterns for each day. I then show comparisons for daily shapes and draw conclusions based on them. The hardware platform used it a T-mobile G1 phone running Android r1.5, using the Asahi Kasei Microdevices AK8967A 3-Axis Electronic Accelerometer.

4.2. G1 Daily Activity Analysis

The purpose of this section is to outline and discuss the data collection process, the work related to activity recognition using an accelerometer, the features used from the accelerometer three axes, classification and finally the applications of E-shape analysis on the work.

4.2.1. Previous Work in Activity Recognition

Over the years there has been many works in the field of activity recognition from accelerometer readings [1, 20, 2, 8, 13]. Most built custom data acquisition boards mounting several single or double axial accelerometers and even fewer used a single triaxial accelerometer mounted in one location. The authors have chosen to use the G1 Android platform for its ease of use and high level of customization. I have also chosen to use the Weka Explorer as a proof-of-concept measurement utility. Using a training set (section 4.2.4, I have successfully replicated the work of [13]. In [13], they used a sliding window of five seconds. The sliding window was broken into several components: mean, standard deviation, energy and correlation. Preliminary analysis using Naive Bayes to train and classify shows an accuracy of 98.56% using 10-Fold cross validation. The confusion matrix can be found at table 4.1.

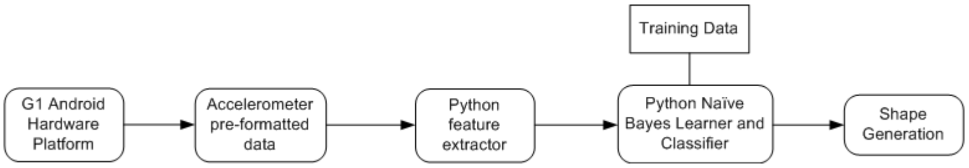
Table 4.1. Confusion matrix for daily activities.

	Sitting	Walking	Downstairs	Upstairs	Running	Driving
Sitting	130	0	0	0	0	5
Walking	0	111	0	0	0	0
Downstairs	0	1	36	0	0	0
Upstairs	0	0	0	35	0	0
Running	0	0	0	0	30	0
Driving	1	0	0	0	0	105

4.2.2. Accelerometer Data Information and Collection

I have developed a general purpose data aquisition program called BlackBox. The application was developed using Android SDK r1.5 and it allows the user to custom log the accelerometer, compass, GPS, or record audio. For this work, the program has been set to record all three axes of the accelerometer for thirty seconds every five minutes over the course of a day. Preliminary data recordings for this paper have been limited to six hours. The BlackBox application can also be set to four different recording frequencies, however a small limitation is that the Android provides the data at a best case scenario, meaning the data stream is not perfectly consistent. There are small variations leading to readings coming in at ten or eleven times per second. The data is then processed through a python script that creates a separate file which is tab formatted with the selected features, see section 4.2.3. A diagram of data flow can be found at Figure 4.1.

Figure 4.1 Data flow diagram for accelerometer readings.


Training Data
G1 Android Hardware Platform
Accelerometer pre-formatted data
Python feature extractor
Python Naïve Bayes Learner and Classifier
Shape Generation

4.2.3. Feature Selection for Daily Activities

A feature selection routine is used to find the best features out of the total twelve that were originally provided. The CFS (Correlationbased Feature Selection algorithm) subset evaluator was chosen from Weka explorer. It provided the best improved results for the Naive Bayes classifier in this instance. Table 4.2 shows the total list of attributes and which were selected for classification. Results from Weka's evaluation are as follows: stale search after 5 node expansions, total number of subsets evaluated is 78 and merit of best subset found is 0.857.

Table 4.2. Total features with selection.

Attribute	X Mean	Y Mean	Z Mean	X Std	Y Std	Z Std
Selected		✓	✓	✓		✓
Attribute	X Energy	Y Energy	Z Energy	XY Corr	XZ Corr	YZ Corr
Selected	✓		✓			✓

4.2.4. G1 Activity Recognition from Training Data

As proof of concept I am continuing the preliminary work using the Weka explorer utility. A set of training data was built from several days worth of measurements. Observed in [13], a persons walk can change slightly from day to day, this will be reflected in the training data set. A sample is given in table 4.3. The unknown sample being classified is walking.

The table shows a 95% accuracy on this particular data sample, it is also the case for the other activities. The table shown is a worst case example, as most of the others samples are classified with 100% accuracy. For determining what context the user is currently in, the mean of the total sampled data is taken. The thirty seconds of recording provides four or five sets for classification. The first activity with greater or equal to 50% recognition is taken for the mean total of the thirty second recording.

Table 4.3. Classification of unknown data from training set.

	Sitting	Walking	Downstairs	Upstairs	Running	Driving
Sitting	0	0	0	0	0	0
Walking	0	19	1	0	0	0
Downstairs	0	0	0	0	0	0
Upstairs	0	0	0	0	0	0
Running	0	0	0	0	0	0
Driving	0	0	0	0	0	0

4.2.5. Python Classification Tool Using Python Orange

The use of Weka explorer was to show a proof of concept for the model I am proposing. Actual implementation is done in Python using a module called Orange [**?**]. This module is a simplified version of the Weka explorer utility. It allows a user to build, train, and use simple Bayesian and Tree based classifiers, as well as options for more advanced users.

The tool takes input from the G1 Blackbox application, which is dozens of accelerometer data files, and outputs the shape for the recorded time period. The script uses a base Naive Bayes learner and classifier. The training data has been converted from Weka ARFF format to be compatible for input with the Orange module which uses tab formatting. The script covers the following steps:

(i) Convert the G1 Accelerometer data into mean, standard deviation, energy and correlation.

(ii) Run the feature selector removing unneeded categories.

(iii) Create a Naive Bayes classifier from the training set.

(iv) Classify the unknown data.

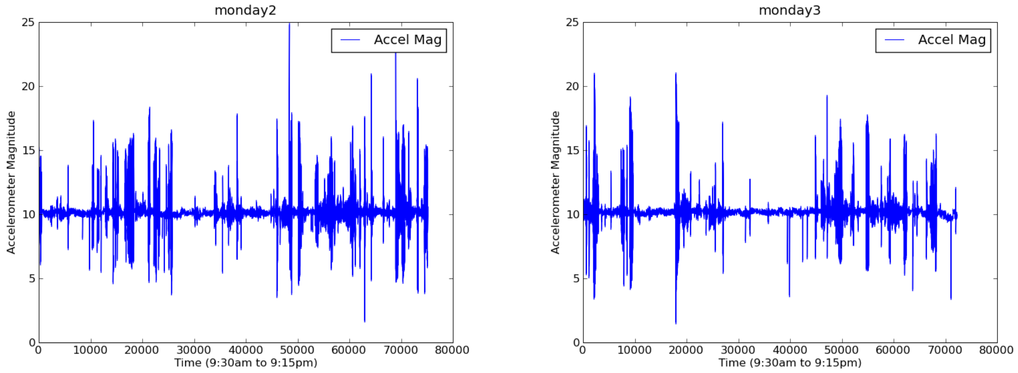
(v) Build a shape string for each recorded period (day).

The shape string is the fed a third python script which is able to store, compare and display all of the data in multiple formats.

4.2.6. E-Shape Analysis as Applied to Daily Activity Recognition

In my previous work I used E-shape analysis in many different methods of email analysis. Here I will apply the same type techniques to daily activities. The purpose is to classify users differing daily shapes. The term daily shape is defined as the mean cumulative activity for each recorded period. The mechanic of my data acquisition is that every five minutes I will record thirty seconds of accelerometer data. In figures 2(a) and 2(b) are couple of the raw samples that are fed into the shape classifier.

Figure 4.2 Side by side of two Monday's raw accelerometer readings.

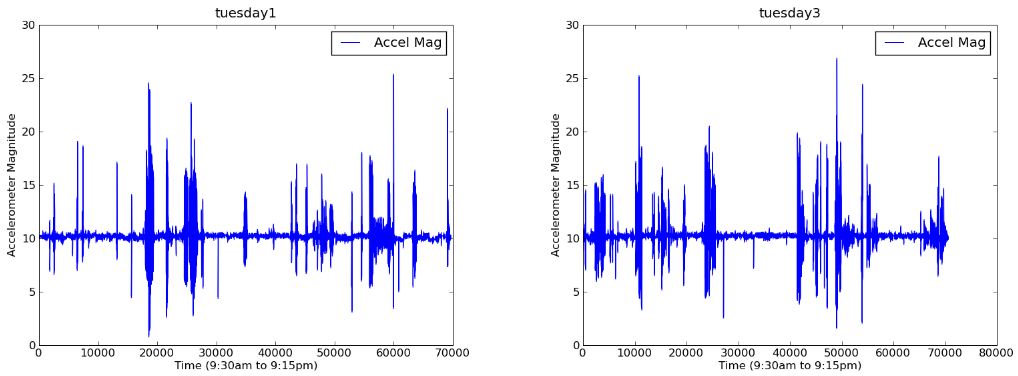


(a) Raw data from Monday 2.

(b) Raw data from Monday 3.

The general patterns observed make it difficult to tell what exactly is going on. Any medium spike or extend spike suggests walking, large spikes are running and/or highly active activity and the remainder is inactive sitting/driving activities. Comparing a Monday day to another Monday day is also a difficult task, however looking at a different day may offer more compelling results. Figures 3(a) and 3(b) are of two Tuesday readings. They provide a slightly different look as the user has class from 2:00pm to 5:00pm on Tuesdays and Thursdays, as observed from the very low activity between readings approximately 30000 and 45000 (2pm to 5pm).

Figure 4.3 Side by side of two Tuesday's raw accelerometer readings.



(a) Raw data from Tuesday 2.

(b) Raw data from Tuesday 3.

To build the shape of a day a context graph is built. Activities are mapped to the context for which they represent and are plotted for each hour of recording. See table 4.4 for context mappings.

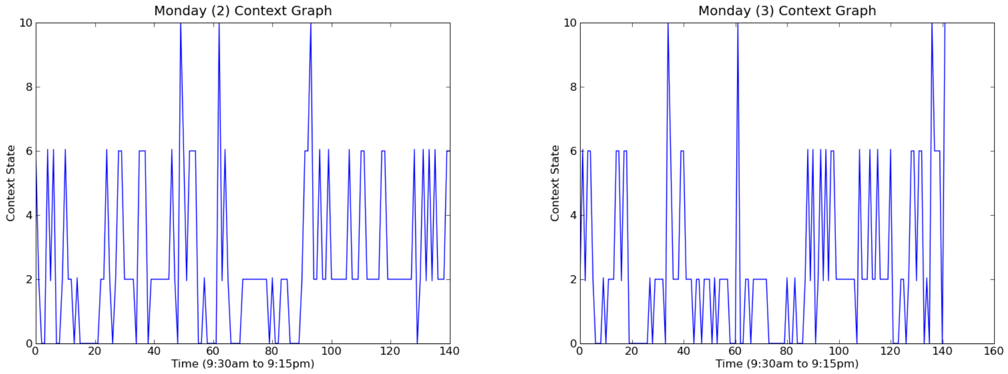
Table 4.4. Context to activity mappings. (Italicized items are for future work.)

Inactive	Active	Very Active	Highly Active
Sitting *Studying at library*	Actively Sitting Working at desk Driving	Walking Stairs	Running *Working Out*

The daily shape plot will have twelve mappings per each hour of recorded time. This plot is considered to be the shape of a day (or over the interval of recording). Continuing the model from the raw data above showing Monday and Tuesday I have the shapes for the corresponding figures. Monday's shape data is found at Figure 4.4, and Tuesday's shape data at Figure 4.5

The inactivity zone between 2:00pm and 5:00pm that was observed in the earlier raw data is no longer visibly present in this shape form. When comparing shapes I use the euclidian distance see in equation 8. The maximum distance for a given comparison is given in equation 9.

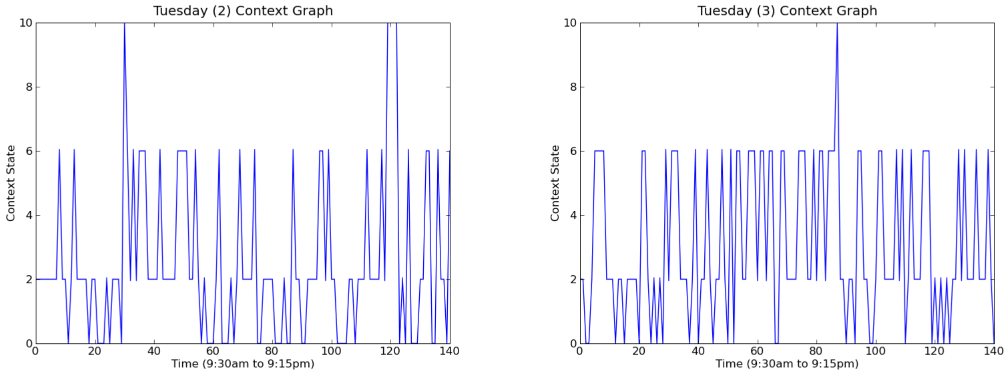
Figure 4.4 Comparison of E-shapes for Monday 2 and 3.



(a) Shape of Monday 2

(b) Shape of Monday 3

Figure 4.5 Comparison of E-shapes for Tuesday 2 and 3.



(a) Shape of Tuesday 2

(b) Shape of Tuesday 3

$$distance(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} \tag{8}$$

$$Maxdistance(p,q) = \sqrt{4 * length(p)} \tag{9}$$

In table 4.5, I show the confusion matrix for the distance of all days. The data being displayed is a sample daily recording of twelve work days for a masters student. The time stamp on all recordings is from 9:30am to 9:15pm. A distance value of zero to two would be close enough to use for individuating different subjects. The subjects behavior is too erratic to provide a specific enough pattern. An example would be if the subject arrived at work, ate lunch, took coffee and restroom breaks with 10seconds of each other day to day, then the shape could provide individuated results. The data does suggest a unique pattern. The seven recorded days are all taken from working at a desk. If he were a retail worker, a driver or any other differing profession, it is believed that it would be reflected in the shape analysis. Further study is currently being conducted to support this hypothesis.

Table 4.5. Confusion Matrix for distance analysis of daily shapes.

	Mon 2	Mon 3	Mon 4	Tues 1	Tues 2	Tues 3	Wed 1	Wed 2	Wed 3	Thur 1	Thur 3	Thur 4
Monday 2	0	40	39	36	38	42	41	36	41	38	41	36
Monday 3	40	0	38	40	37	43	41	36	40	35	41	38
Monday 4	39	38	0	37	41	41	41	36	43	38	40	38
Tuesday 1	36	40	37	0	39	42	43	36	41	38	39	38
Tuesday 2	38	37	41	39	0	40	40	34	37	34	37	38
Tuesday 3	42	43	41	42	40	0	39	37	44	39	42	39
Wednesday 1	41	41	41	43	40	39	0	34	39	37	39	40
Wednesday 2	36	36	36	36	34	37	34	0	35	33	33	36
Wednesday 3	41	40	43	41	37	44	39	35	0	37	39	40
Thursday 1	38	35	38	38	34	39	37	33	37	0	37	39
Thursday 3	41	41	40	39	37	42	39	33	39	37	0	39
Thursday 4	36	38	38	38	38	39	40	36	40	39	39	0

4.3. G1 Individual Walking Analysis

This section covers the individual walking patterns of four different test subjects. In a similar progression to the above work, E-shape analysis will be used to map the individual walking patterns of subjects. Data was collected for four minutes of walking the hallways at UNT discovery park. Each subject wore a pouch to securely fasten the phone to their hip.

The phone orientation was consistent for each subject. The same components and sliding windows are used from above [13].

A Naive Bayes classifier was used along with a 3 fold cross validation. The accuracy was 99.33%. A confusion matrix is located at table 4.6.

Table 4.6. Confusion matrix for individual walking analysis.

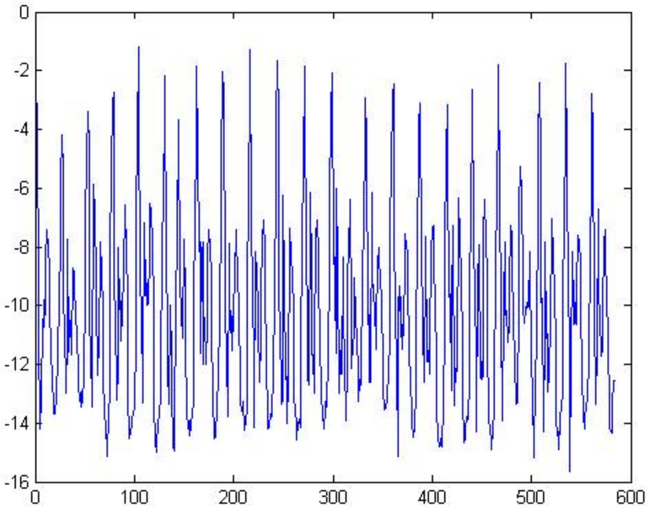
	Subject 1	Subject 2	Subject 3	Subject 4
Subject 1	74	0	2	0
Subject 2	0	76	0	0
Subject 3	0	0	77	0
Subject 4	0	0	0	73

4.3.1. E-Shape Analysis

The individual walking patterns are easily classified using similar technique to the activity recognition. But, how does one person's walk differentiate to another? Is the person left handed or right handed, are they male or female, how much does age play a role in your individual walking pattern? Using E-shape analysis I plan to show the difference in walking patterns for subjects.

A single subjects walking pattern can be seen in Figure 4.6. The differences in a persons pattern can be observed through the periodic signal, amplitude, energy, and correlation. I plan to collect data relating to a wide range of people and show that certain aspects of the accelerometer signal can be used to differentiate the different components of who a person is.

Figure 4.6 Walking Signal for approximately forty-five seconds.



CHAPTER 5

CONCLUSIONS

In this thesis I defined E-shape as the general concept of quantifying events into similar patterns. Then, I show three papers on the applications of E-Shape. They were email analysis, activity detection and daily activity detection. E-Shape is a powerful way to look at data by taking content and context out of the classification process, it allows the classifier to get a "human look" at data to offer side classification.

The first of the papers covers E-Shape as applied to email research. I looked at using E-Shape to understand the detection of botnet's and botnet emailing templates. It is believed that botnet spammer's couldn't change an email for every sent email, that would mean billions of unique emails per day, but rather they use a single template to beat spam filters. A new template could be used to get thousands of emails past a filter before the filter learns that it is spam. However, E-Shape doesn't depend so heavily on content and would allow it to catch botnet behavior much sooner. In my experiments I used a 1200 email corpus including four different languages and five different size domains. Using my E-Shape method of classification I show an overall accuracy of 81%.

Next, after having such successful results using E-Shape on botnet detection, I looked at using it to classify ham and spam emails. Currently spam filters are hugely successful. They offer success rates of over 99% with minimal false positives. However, the power of E-Shape comes in when use it on spam filtering, while not reading any content I am able to show an accuracy of almost 70%. While at first glance this may not look that great, I am not trying to compete with the already successful spam filters, but rather compliment its decision engine with a non-content, non-context analysis of an email. In this experiment I use the TREC 2007 corpus.

Lastly, in my work with email, I look at the personal behavior of users. This work is in its preliminary stages and I simply turn my algorithm on to a users inbox. My corpus is the top three receivers to a single "everyday" inbox. I then run the E-Shape engine with the botnet detection flags on and show that the three senders can be classified with a 75% accuracy. Of note in this experiment, one of the senders was a weekly online newsletter which was classified at 100% accuracy with no false positives.

As I expanded my work in E-Shape I moved into the behavioral based activity detection and how E-Shape could apply. My experiments included taking accelerometer recordings of a subject using the T-Mobile G1 Android platform. These experiments were broken into two sections, pertaining to activity recognition and to daily activity recognition. In the first I go over the Google Android platform and accelerometer readings and analyze how closely they are related to each other with a series of covariance and correlation techniques. In the second I use a paper [13] as the basis of my look into daily activities and E-Shape analysis thereof. I find that users behavior based on raw data could be classified but I am still work on the model to bring it all together. In this section, I show experimental results using euclidian distance based context graphs.

In this thesis I have shown the beginning of E-Shape analysis and how it applies to two area's of study: email and context behavior. I have shown how context and content free analysis can be a powerful tool standing alone but also as a complimentary method to existing techniques.

BIBLIOGRAPHY

[1] Jonghun Baek, Geehyuk Lee, Wonbae Park, and Byoung-Ju Yun, *Accelerometer signal processing for user activity detection*, vol. 3215/2004, pp. 610–617, Springer Berlin/Heidelberg, 2004.

[2] Ling Bao and Stephen S. Intille, *Activity recognition from user-annotated acceleration data*, Pervasive 2004 (2004), 1–17.

[3] Mark Brownlow, *Email and webmail statistics*, April 2008.

[4] L. Le Cam and G. L. Yang, *Asymptotics in statistics: Some basic concepts*, Springer-Verlag, 2000.

[5] Gordon V. Cormack, *Email spam filtering: A systematic review*, Found. Trends Inf. Retr. 1 (2007), no. 4, 335–455.

[6] Gordon V. Cormack, José María Gómez Hidalgo, and Enrique Puertas Sánz, *Spam filtering for short messages*, CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management (New York, NY, USA), ACM, 2007, pp. 313–320.

[7] Gordon V. Cormack and Thomas R. Lynam, *TREC 2007 Public Corpus*, 2007.

[8] MJ Mathie, AC Coster, NH Lovell, and BG Celler, *Detection of daily physical activities using a triaxial accelerometer.*, Med Biol Eng Comput 41 (2003), no. 3, 296–301 (eng).

[9] Emanuel Parzen, *On Estimation of a Probability Density Function and Mode*, The Annals of Mathematical Statistics 33 (1962), no. 3, 1065–1076.

[10] Ryan Paul, *Researchers track Ron Paul spam back to Reactor botnet*, December 2007.

[11] Anirudh Ramachandran and Nick Feamster, *Understanding the network-level behavior of spammers*, SIGCOMM '06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications (New York, NY, USA), ACM, 2006, pp. 291–302.

[12] Anirudh Ramachandran, Nick Feamster, and David Dagon, *Detecting botnet membership with dnsbl counterintelligence.*, Botnet Detection (Wenke Lee, Cliff Wang, and David Dagon, eds.), Advances in Information Security, vol. 36, Springer, 2008, pp. 131–142.

[13] Nishkam Ravi, Nikhil Dandekar, Prreetham Mysore, and Michael L. Littman, *Activity recognition from accelerometer data*, American Association for Artificial Intelligence (2005).

[14] S. J. Sheather and M. C. Jones, *A reliable data-based bandwidth selection method for kernel density estimation*, Journal of the Royal Statistical Society, Series B (1991), no. 53, 683–690.

[15] Sara Sinclair, *Adapting bayesian statistical spam filters to the server side*, J. Comput. Small Coll. 19 (2004), no. 5, 344–346.

[16] Joe Stewart, *Top Spam Botnets Exposed*, April 2008.

[17] Salvatore J. Stolfo, Shlomo Hershkop, Chia-Wei Hu, Wei-Jen Li, Olivier Nimeskern, and Ke Wang, *Behavior-based modeling and its application to email analysis*, ACM Trans. Internet Technol. 6 (2006), no. 2, 187–221.

[18] W. T. Strayer, D. Lapsley, R. Walsh, and C. Livadas, *Botnet detection based on network behavior*, Botnet Detection: Countering the Largest Security Threat (Wenke Lee, Cliff Wang, and David Dagon, eds.), Springer-Verlag, 2007.

[19] TRACElabs, *Template Based Spam*, May 2009.

[20] P.H. Veltink, HansB.J. Bussmann, W. de Vries, WimL.J. Martens, and R.C. Van Lummel, *Detection of static and dynamic activities using uniaxial accelerometers*, Rehabilitation Engineering, IEEE Transactions on 4 (1996), no. 4, 375–385.

[21] Chih-Ping Wei, Hsueh-Ching Chen, and Tsang-Hsiang Cheng, *Effective spam filtering: A single-class learning and ensemble approach*, Decis. Support Syst. 45 (2008), no. 3, 491–503.