A SMARTPHONE-BASED SYSTEM FOR AUTOMATED DETECTION OF WALKING

FINAL PROJECT REPORT

by

Philip M. Hurvitz
University of Washington

Sponsorship
PacTrans
University of Washington Urban Form Lab

for
Pacific Northwest Transportation Consortium (PacTrans)
USDOT University Transportation Center for Federal Region 10
University of Washington
More Hall 112, Box 352700
Seattle, WA 98195-2700

In cooperation with US Department of Transportation-Research and Innovative Technology Administration (RITA)

Disclaimer

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the U.S. Department of Transportation's University Transportation Centers Program, in the interest of information exchange. The Pacific Northwest Transportation Consortium, the U.S. Government and matching sponsor assume no liability for the contents or use thereof.

Technical Report Documentation Page

1. Report No. 2013-S-UW-0051	**2. Government Accession No.**	**3. Recipient's Catalog No.**
4. Title and Subtitle Smartphone-Based System for Automated Detection of Walking – year 2 (2013-14)		**5. Report Date** 8/31/2015 **6. Performing Organization Code**
7. Author(s) Philip Hurvitz		**8. Performing Organization Report No.** 51-626672
9. Performing Organization Name and Address PacTrans Pacific Northwest Transportation Consortium University Transportation Center for Region 10 University of Washington More Hall 112 Seattle, WA 98195-2700		**10. Work Unit No. (TRAIS)** **11. Contract or Grant No.** DTRT12-UTC10
12. Sponsoring Organization Name and Address United States of America Department of Transportation Research and Innovative Technology Administration		**13. Type of Report and Period Covered** 9/30/2013-8/31/2015 **14. Sponsoring Agency Code**
15. Supplementary Notes Report uploaded at www.pacTrans.org		

16. Abstract

Walking is the most effective mode of travel to access transit: transit hubs with higher residential and employment densities have higher ridership levels because they serve areas where a large population is within a short walk of transit service. Walking has additional benefits: it is well-known as a low impact mode of travel for short trips to and from, as well as within, commercial areas; and it is the most popular form of physical activity. However, current data on walking are notoriously poor. Travel surveys and diaries underestimate walking activity and lack information on walking paths taken, thereby undermining transportation policies that can encourage sustainable travel. Objective data on how often, how long and where people walk are essential to support environmentally friendly and safe transportation systems.

17. Key Words Data and Information Technology; Pedestrians and Bicyclists; Planning and Forecasting; Public Transportation		**18. Distribution Statement** No restrictions.	
19. Security Classification (of this report) Unclassified.	**20. Security Classification (of this page)** Unclassified.	**21. No. of Pages**	**22. Price** NA

Form DOT F 1700.7 (8-72) **Reproduction of completed page authorized**

Table of Contents

List of Figures

List of Tables

Acknowledgments

The PI wishes to thank Sep Makhsous, Dhruv Malik, and Nikhil Pailoor, the UW Electrical Engineering students who worked on technical aspects of the project. Professor Anne Vernez-Moudon of the UW College of Built Environments Department of Urban Design and Planning provided matching funds.

Executive Summary

Walking is the most effective mode of travel to access transit: transit hubs with higher residential and employment densities have higher ridership levels because they serve areas where a large population is within a short walk of transit service (1). Walking has additional benefits: it is well-known as a low impact mode of travel for short trips to and from, as well as within, commercial areas; and it is the most popular form of physical activity (2,3).

Despite the importance of walking, current data on walking are notoriously poor. Travel surveys and diaries underestimate walking activity (4) and lack information on walking paths taken, thereby undermining transportation policies that can encourage sustainable travel. Objective data on how often, how long and where people walk are essential to support environmentally friendly and safe transportation systems.

This study made measurements with a combination of GPS, accelerometry, and high-frequency photos, tracking 12 free-roaming individuals during waking hours over the course of a three-day period. Measurements were made at a 10 s interval, with photos taken using a neck-lanyard mounted smartphone. A deterministic classifier was first used to separate walking and non-walking physical activity bouts, following Kang et al. (4).

Point-of-view images obtained from the smartphones during physical activity bouts failed automated image processing techniques to support the main study aims. Due to image blur and rapidly changing point of view, successive images failed multiple image registration methods. Without registration, it was not possible to detect frame-to-frame similarities and differences (i.e., all successive images from physical activity bouts were completely different based on automated image processing techniques). Smartphone cameras to support the main

study aims would require substantial image stabilization (i.e., a gimbal), making them impractical for ubiquitous sensing.

Despite the failure of the main aim, some promising work was accomplished in development of algorithms for (1) differentiating physical activity types based on accelerometry data and (2) differentiating bicycling from other activities based on single image frames, both using Naïve Bayes classifiers.

Major findings and their implications

- Use of neck-lanyard mounted smartphones for photographic image capture to support detection of walking produces images not able to be registered using automated image processing techniques
- Single image frames can be used to differentiate bicycling from other activities using Naïve Bayes classifiers; with additional work this framework could be extended to other activities as well
- Classification of physical activity types using Naïve Bayes classifiers applied to accelerometry data is promising

Chapter 1. Introduction

1.1 Problem Statement

Walking is the most effective mode of travel to access transit: transit hubs with higher residential and employment densities have higher ridership levels because they serve areas where a large population is within a short walk of transit service (1). Walking has additional benefits: it is well-known as a low impact mode of travel for short trips to and from, as well as within, commercial areas; and it is the most popular form of physical activity (2,3). However, current data on walking are notoriously poor. Travel surveys and diaries underestimate walking activity (4) and lack information on walking paths taken, thereby undermining transportation policies that can encourage sustainable travel. Valid data on how often, how long and where people walk are essential to support environmentally friendly and safe transportation systems.

1.2 Objective

The objective of this study was use a combination of high resolution global positioning system (GPS), accelerometer, and point-of-view photograph data for classifying activity into walking and non-walking intervals. Current methods for identifying walking episodes using only GPS and accelerometry are promising, but without a 'gold standard' for validation, it is not possible to determine whether such methods are accurate. In this study, photos taken at 10 s intervals from a neck-lanyard-mounted smartphone were used to attempt enhancement of classification of walking and non-walking episodes, as well as to provide a validation data set for interactive review.

1.3 Overview

This report begins with a brief literature review, which is followed by methods, and results. An appendix provides supplemental material, including recruitment documents, participant manuals, and scripts used in data preparation, management, and analysis.

Chapter 2. Literature Review

A growing body of research has focused on the use of accelerometry and GPS as a way to estimate the time and location in which physical activity occurs (5–8). However, there are no robust and validated methods of processing these data for identifying walking as a specific type of physical activity. The PI's research group at the Urban Form Lab (UFL) is currently working on the cutting edge of measurement and analysis of walking using multiple data sources. We use a hybrid of GPS, accelerometry, and travel diary data linked by common timestamp, with potential walking bouts identified by intervals with relatively rectilinear trajectories, speeds between 2 and 6 kmh^{-1}, and accelerometry values between 1000 and 5725 counts per 1 s epoch, and augmented by diary entries (4).

A limitation of this method is the reliance on plausible tolerances from GPS units and accelerometers, with a lack of "gold standard" objective data. A second limitation is the use of travel diaries, lowering the overall reliability of identification of walking bouts. Photographs taken at regular intervals from the perspective of free-roaming individuals may provide such an additional data source for increasing the confidence in differentiating walking from other activities. Like accelerometry and other sensors available in the Smartphone, photos can be taken consistently without the frequent loss of signal common for GPS data.

Underscoring this line of research, a recent issue of the *American Journal of Preventive Medicine* focused on current state-of-the-art use of neck-lanyard mounted cameras (the Microsoft SenseCam (9) and generic Smartphones) in lifestyle-related health research. Applications spanned dietary assessment (10), memory and cognition (11), and classification of activity (12).

Advanced image processing techniques allow for automated registration (13), commonly used in the field of medical tomography (14) and satellite remote sensing (15). These techniques could be applied to point-of-view images taken from free roaming individuals using body-mounted smartphones. Once registered, frame-to-frame differences could be used to quantify the similarity of image sequences (16). The primary aim of the current study is to test whether this measurement and processing framework could be used to differentiate walking and non-walking episodes using a combination of GPS, accelerometry, and photographic data.

Chapter 3. Methods

3.1 Sample

A convenience was obtained using targeted e-mail messages and word-of-mouth. Subjects were screened for eligibility based on the following characteristics: age (between 18 and 65 y, or between 15 and 65 y if a UW student); English proficiency; ability to walk 20 min; being a habitual walker (walking ≥ 15 m/d on a typical day); able to perform daily routines without assistance or special equipment; not having medical conditions contraindicating exercise (e.g., cardiovascular disease, high blood pressure). Subjects were consented, instructed in use of devices, measured for weight and height, and completed a brief sociodemographic survey. Upon completion of the three-day study period and return of the devices, subjects were paid $20 as partial compensation for their time and efforts. Study procedures were approved by the University of Washington Human Subjects Division (application 47065).

3.2 Data collection

3.2.1 GPS

GPS data were collected using a Qstarz BG-Q1000XT (Qstarz, Taipei, Taiwan) data logger set to record at an interval of 10 s. Devices were configured to record time (UTC seconds), location (latitude, longitude), height (altitude of a place above sea level or ground level in m), speed in km/h, heading (the compass direction of the longitudinal axis of a motion), PDOP (positional dilution of precision = position accuracy of 3d-coordinates), HDOP (horizontal dilution of precision = horizontal accuracy of 2d-coordinates; VDOP (vertical dilution of precision = vertical accuracy of height), number of satellites (number used for fix and number in view), satellite id ,satellite elevation, satellite azimuth (angle from horizon), satellite signal to noise ratio, and distance between successively logged locations in m. The GPS could be carried in a pocket, purse, backpack, etc. or worn on a belt (Figure 3-1).

3.2.2 Accelerometry

Three-dimensional accelerometry data were collected with an Actigraph GT3M (Actigraph Corp., Pensacola, FL), also set to collect aggregated count data at a 10 s epoch duration. The accelerometer was mounted on a belt, to be worn at the right iliac crest (Figure 3-1). GPS and accelerometry data were combined using common time stamps and saved as PostgreSQL (17) tables, with points stored as PostGIS (18) geometry features for spatial analysis.

Figure 3-1: Position of accelerometer and GPS

3.2.3 Photographs

Photos were taken using a ZTE Warp 4G Android phone worn in a custom-made pouch on a neck lanyard (Figure 3-2). The “Dailyroads Voyager” app (http://www.dailyroads.com/voyager.php) was used to capture images at a 10 s interval, at a resolution of 2592x1944 pixels.

Figure 3-2: Smartphone position

3.3 Analysis

3.3.1 Parsing accelerometry into 'bouts' of physical activity

Raw accelerometry data were first processed to remove extended periods of wearing and nonwearing. A nonwearing interval was identified as having zero counts for one hour or more (allowing for one minute of nonzero counts within the hour). "Valid" days of accelerometry were those having at least one hour of wearing data.

Within wearing intervals, a "bout" of physical activity consistent with walking was defined as having sustained counts between 2,000 and 6,166 counts per 1 minute epoch, consistent with physical activity levels associated with walking (19). The minimum interval for a physical activity bout was set to 5 min, or a minimum of 7 min including 2 min of counts outside this threshold.

3.3.2 Initial walking and non-walking classification

"Walking" was defined as human-powered and non-mechanical (i.e., not cycling) movement through space at a sustained pace and physical activity level. Physical activity bouts were classified as walking or non-walking based on thresholds of speed (mean between 2 and 6 km/h), GPS coverage (at least 20% of bout records with GPS), and spatial clustering (Figure 3-3).

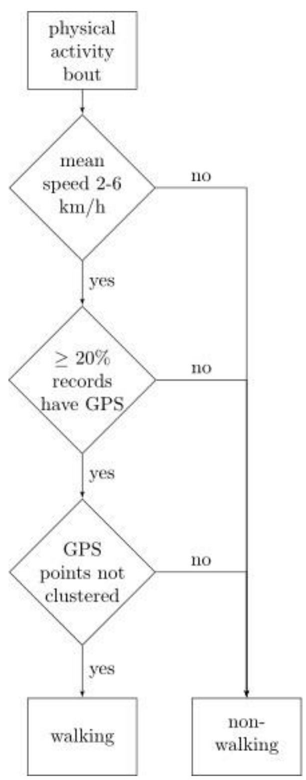

Figure 3-3: Decision tree for differentiating walking and nonwalking based on GPS characteristics

To operationalize clustering, the distance between each possible pair of GPS points was measured; those points below the 95th percentile of distances were used to create a minimum bounding circle; if that circle's radius was ≤ 20 m, the physical activity bout was determined to have taken place at a single location and was therefore not walking.

3.3.3 Image registration

A number of different automated image registration software packages were used for registering serial images, as shown in Table 3-1.

Table 3-1: Image registration software

software	source
AIR	http://bishopw.loni.ucla.edu/air5/
SimpleITK	http://www.itk.org/
SimpleElastix	https://simpleelastix.readthedocs.org/
imreg.py	http://www.lfd.uci.edu/~gohlke/code/imreg.py.html
image_registration 0.2.1	https://pypi.python.org/pypi/image_registration/0.2.1
pyimreg	https://github.com/pyimreg
Opticks	http://opticks.org/

Due to the nature of the paired image geometries, the *affine* transformation was used, which allows for translation, rotation, shearing, and scaling.

3.3.4 Physical activity type classification using Naïve Bayes classifiers

To differentiate among different types of physical activity (walk, run, sit, stand, other) based on accelerometry, a Naïve Bayes classifier was developed. The first pass dichotomizes moving and nonmoving based on total acceleration. It should be noted that certain types of physical activity are not well suited to measurement through accelerometry, such as bicycling (20,21). The moving activities were first sorted in to walking and running classes. A second classifier was developed for sorting into walking vs. other and running vs. other. Finally, nonmoving activities were sorted in to sitting and standing based on acceleration patterns directly preceding the non-moving interval. A sample of 37 walking, 23 running, 15 sitting, and 24 standing, episodes were measured to test the respective algorithms.

3.3.5 Single-frame classification of bicycling vs. other activities

Because of the problem of automatically identifying bicycling from accelerometry data, an image-based classifier for single frames was developed. The classifier used standard image smoothing, enhancement, and high-contrast convolution filters. The smoothing filter used a 9 x 9 pixel kernel, assigning the mean of the pixels in the kernel to the central pixel.

Edges were then consolidated into coordinate data and sorted using a Naïve Bayes algorithm. An ArrayList was created, composed of (1) high-intensity pixels marked as TRUE, and (2) XY coordinates. The bicycle's handlebars and frame, from a set of 20 training reference images, provided a prominent and common set of edges for inter-image comparison, and another 20 random images were sampled from known walking episodes. Test images were processed using the same convolution and ArrayList construction, and compared to reference data from a known bicycling image using a Gaussian Naïve Bayes algorithm. Above a threshold of similarity, the test image was classified as non-walking. A more detailed report of this image classification is provided in 0, and Java code for bicycling image classification is available in 0.

Chapter 4. Results

4.1 Sample

Twelve participants were recruited. Measurements spanned August, 2014 to February, 2015. Table 4-1 shows demographic characteristics of the sample, which was relatively young (mean age 30.1 y), predominantly male, and highly educated (5 graduate degrees). Household incomes where bimodally distributed, with about ½ of the participants earning <$40k/y and the remainder earning >$70k/y, and relatively few were home owners. The majority were full- or part-time employed, and 4 were students. Half of the participants lived in multi-family housing. Most participants had driver's licenses and had access to a car. Over half were married. The sample was relatively lean, with a mean BMI of 26.0. Although 12 subjects were enrolled, one did not complete the survey.

Table 4-1: Participant characteristics

total, n (took survey)	12 (11)
age, mean (sd)	30.1 (10.6)
male, n	9
education, n	
some college	5
completed college	1
completed graduate degree	5
household income, n	
<10k	4
20-30k	1
40-50k	1
70-80k	2
90-100k	1
>100k	1
employment status, n (1 missing)	
full-time	6
part-time	3
unemployed	1
student, n	4

dwelling type, n (1 missing)	
single-family	5
multi-family	5
household size, mean (sd)	2.7 (1.2)
home owners, n	3
children in household, mean (sd)	0.5 (0.8)
n cars in hh, mean (sd)	0.73 (0.9)
has driver's license, n	8
married, n	6
BMI, kg/m^2, mean (sd)	26.0 (9.4)

4.2 Accelerometry data

A total of 153,418 accelerometry records were collected over 52 valid person-days, with a mean of 4.3 days per subject (SD 1.0). The mean duration of 'wearing' accelerometry was 8.3 hours per day (SD 1.8)

4.3 GPS data

A total of 148,463 GPS records were collected over 47 person-days, corresponding to a mean of 3158.7 locations per person-day, or 8.8 h per person-day assuming uninterrupted recording at a 10 s interval.

4.4 Photos

A total of 49,261 images were captured, for a mean of 960.1 per person-day, corresponding to 2.7 h of images per person day. Capturing images at a 10 s interval resulted in very short battery life. Subjects reported the phone getting quite warm during data collection.

4.5 Physical activity and walking bouts

Using only accelerometry and GPS data, there were a total of 109 physical activity bouts, (using a 5 minute minimum duration and accelerometry between 2,000 and 6,166 counts per min) for 10 subjects (two subjects did not have any physical activity bouts), for an average of 9.9 bouts/subject (sd = 6.6).

Of these 109 bouts, 69 met, and 40 did not meet the walking criteria outlined in Figure 3-3. The mean duration of a bout was 11.7 min (SD = 5.9), as shown in Table 4-2. The mean duration and variation of non-walking and walking bouts was quite similar.

Table 4-2: Physical activity bout durations

type	n bouts	mean (min)	sd (min)
all	109	11.7	5.9
walk	69	11.7	5.7
non-walk	40	11.6	6.4

4.6 Image registration

Each tested application was able to register sample images provided in the distribution packages, and some were able to register images from sedentary periods during which the camera was relatively steady (Figure 4-1). This example, processed with SimpleElastix, returned a greyscale version of the registered image.

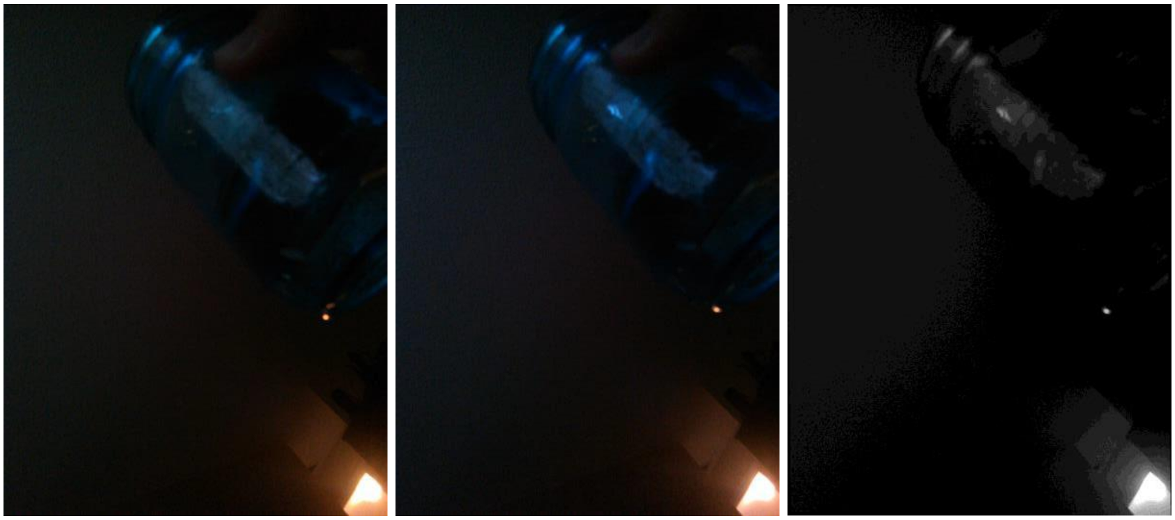

Figure 4-1: Images from sedentary activity (left = “target”, center = “moving”) and registered (right)

A typical series of photos extracted from a physical activity bout that was classified as walking based on GPS and accelerometry is shown in Figure 4-2. This sequence was selected as an example of a common street scene, and with some features in common (the tree to the left of the frame, telephone poles and wires). However, after automatic registration, the “moving” image was unable to be well registered to the “target” image, and returned a copy of the “moving” image.

Figure 4-2: Images from GPS and accelerometry detected walking activity (far left = “target”, center left = “moving”), registered (center right) and registered composed with target (far right)

Some slight image shifting is seen at the bottom of the composite of the fixed, moving, and registered images (Figure 4-3). However, it appears that there was insufficient capability to generate tie points between the fixed and moving image.

Figure 4-3: Composite of moving and registered image

Pixel differences between the target and registered image were substantial enough to have essentially no “shared” features or regions. This was the case for all images taken during physical activity bout intervals.

4.7 Differentiation of physical activity types using accelerometry

Of the 99 activity episodes, 82 (82.8%) were accurately classified, as shown in Table 4-3.

Table 4-3: Results of physical activity type classification

activity	episodes classified	correct classification	% correctly classified
walking	37	32	86.5
running	23	20	87.0
sitting	15	10	66.7
standing	24	20	83.3

4.8 Image classification: bicycling vs. other

Classification of 'bicycling' vs. 'other' activities gave some promising results. Based on the test sample of images, bicycling was correctly classified for 62% of images.

Chapter 5. Discussion

This study attempted to use intervalometric point-of-view photos taken with a neck-lanyard mounted smartphone to augment a GPS and accelerometry based algorithm for differentiating physical activity episodes as ‘walking’ and ‘non-walking.’ Image registration techniques were attempted, which largely failed to produce usable results for photos taken during physical activity bouts.

In the two fields in which automated image registration is used, medical tomography, and remote sensing, there is a relatively high degree of control of image framing. For clinical tomography, such as CAT, MRI, or PET scanning, subjects are placed in the imaging device according to careful specifications, which result in image series with similar geometries. Likewise, remotely sensed images are typically captured from satellites or fixed-wing aircraft at altitudes high enough for landscape features to have similar geometries across successive frames. Furthermore, photogrammetric cameras and satellite sensors are specially engineered for handling this type of image processing (e.g., known focal length, fiducial markers, filters or sensors for specific wavelengths). And in both of these fields, images are taken under carefully controlled conditions to reduce blur, widely differing camera angles, and changes in distance from camera to subject.

Due to the inability to meet the fundamental requirement of image registration, we were therefore unable to meet the objective of quantifying image differences across successive frames. An image capturing system with better stability would likely be needed. New generation cameras such as the GoPro Hero4 Session, specifically designed for capturing video during active sports are likely to function far better than low-to-moderate end Android phones. However, this new generation camera is able to capture only up to two hours of video on a single charging cycle, making it impractical for use in studies of free-roaming individuals.

Despite the failure to achieve the primary aim, some promising work was performed for using accelerometry data alone for differentiating among a number of types of physical activity. Further work in applying Naïve Bayes classifiers to these data, perhaps augmented with other sensor data available in newer generation smartphones or other wearable sensors may prove fruitful. Additionally, some progress was made in differentiating cycling from other activities by using Naïve Bayes classifiers on images taken with the point-of-view camera. Because bicycling episodes are difficult to capture using only accelerometry, applying this method to data collected by multiple sensors may aid in estimation of the duration spent cycling as part of the entire duration of physical activity.

Chapter 6. Conclusions and Recommendations

Based on the pilot research outlined in this report, we have some final conclusions and recommendations.

6.1 Use of smartphones for capturing point-of-view imagery

The smartphone ended up not being a very useful tool for capturing high-frequency point-of-view photographic data for generating automatically registered sequences. The main reasons were blur from camera motion and the exceedingly different image frame in successive images. Smartphones with advanced image stabilization may provide better imagery. Other dedicated cameras (e.g., various GoPro models) may also perform better, although they would suffer the same limited battery life problem as the smartphones we tested.

6.2 Activity classification using accelerometry

Applying Naïve Bayes classification to accelerometry data for differentiating activity types is promising. Additional controlled trials using a range of activities, as well as sensitivity analyses for run-time parameters are likely to greatly improve classification success.

6.3 Technology transfer

At this time, further research and development would be required before the work presented in this report could result in widely used or marketable software algorithms.

References

1. Dunphy R, Cervero R, Dock F, McAvey M, Porter D, Swenson C. Developing around transit [Internet]. Washington, DC: Urban Land Institute; 2005 [cited 2013 May 16]. Available from: http://books.google.com/books/about/Developing_around_transit.html?id=r9u3AAAAIAAJ

2. Hu FB, Stampfer MJ, Solomon C, Liu S, Colditz GA, Speizer FE, et al. Physical Activity and Risk for Cardiovascular Events in Diabetic Women. Ann Intern Med. 2011;134:96–105.

3. Manson JE, Hu FB, Rich-Edwards JW, Colditz G a, Stampfer MJ, Willett WC, et al. A prospective study of walking as compared with vigorous exercise in the prevention of coronary heart disease in women. N Engl J Med [Internet]. 1999 Aug 26;341(9):650–8. Available from: http://www.ncbi.nlm.nih.gov/pubmed/10460816

4. Kang B, Moudon AV, Hurvitz PM, Reichley L, Saelens BE. Walking objectively measured: Classifying accelerometer data with GPS and travel diaries. Med Sci Sports Exerc [Internet]. 2013 Jul [cited 2013 Jun 23];45(7):1419–28. Available from: http://www.ncbi.nlm.nih.gov/pubmed/23439414

5. Troped PJ, Wilson JS, Matthews CE, Cromley EK, Melly SJ. The Built Environment and Location-Based Physical Activity. Am J Prev Med. 2010;38(4):429–38.

6. Rodríguez DA, Aytur S, Forsyth A, Oakes JM, Clifton KJ. Relation of modifiable neighborhood attributes to walking. Prev Med (Baltim). 2008 Sep;47(3):260–4.

7. Rodríguez DA, Cho G-H, Elder JP, Conway TL, Evenson KR, Ghosh-Dastidar B, et al. Identifying Walking Trips From GPS and Accelerometer Data in Adolescent Females. J Phys Act Health [Internet]. 2012 Mar [cited 2012 Apr 12];9(3):421–31. Available from: http://www.ncbi.nlm.nih.gov/pubmed/21934163

8. Rodríguez DA, Cho G-H, Evenson KR, Conway TL, Cohen DA, Ghosh-Dastidar B, et al. Out and about: association of the built environment with physical activity behaviors of adolescent females. Health Place [Internet]. 2012 Jan [cited 2012 Mar 26];18(1):55–62. Available from: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3259163&tool=pmcentrez&rendertype=abstract

9. Microsoft Research. Microsoft Research SenseCam [Internet]. [cited 2013 May 16]. Available from: http://research.microsoft.com/en-us/um/cambridge/projects/sensecam/

10. O’Loughlin G, Cullen SJ, McGoldrick A, O’Connor S, Blain R, O’Malley S, et al. Using a wearable camera to increase the accuracy of dietary analysis. Am J Prev Med [Internet]. Elsevier Inc.; 2013 Mar [cited 2013 Feb 20];44(3):297–301. Available from: http://www.ncbi.nlm.nih.gov/pubmed/23415128

11. Silva AR, Pinho S, Macedo LM, Moulin CJ. Benefits of SenseCam Review on Neuropsychological Test Performance. Am J Prev Med [Internet]. Elsevier Inc.; 2013 Mar [cited 2013 Feb 20];44(3):302–7. Available from: http://www.ncbi.nlm.nih.gov/pubmed/23415129

12. Kerr J, Marshall SJ, Godbole S, Chen J, Legge A, Doherty AR, et al. Using the SenseCam to Improve Classifications of Sedentary Behavior in Free-Living Settings. Am J Prev Med [Internet]. Elsevier Inc.; 2013 Mar [cited 2013 Feb 20];44(3):290–6. Available from: http://www.ncbi.nlm.nih.gov/pubmed/23415127

13. Zitová B, Flusser J. Image registration methods: A survey. Image Vis Comput. 2003;21(11):977–1000.

14. Maintz JBA, Viergever M a. A survey of medical image registration METHODS. 1998;2(1):1–36.

15. Kennedy RE, Cohen WB. Automated designation of tie-points for image-to-image coregistration. Int J Remote Sens. 2003;24(17):3467–90.

16. Radke RJ, Andra S, Al-Kofahi O, Roysam B. Image change detection algorithms: a systematic survey. IEEE Trans Image Process [Internet]. 2005 Mar;14(3):294–307. Available from: http://www.ncbi.nlm.nih.gov/pubmed/15762326

17. The PostgreSQL Global Development Group. PostgreSQL.

18. The PostGIS Development Group. PostGIS. 2008.

19. Sasaki JE, John D, Freedson PS. Validation and comparison of ActiGraph activity monitors. J Sci Med Sport [Internet]. Sports Medicine Australia; 2011;14(5):411–6. Available from: http://dx.doi.org/10.1016/j.jsams.2011.04.003

20. Crouter SE, Clowers KG, Bassett DR, Scott E, Jr a. A novel method for using accelerometer data to predict energy expenditure. 2006;1324–31.

21. Ermes M, Pärkkä J, Mäntyjärvi J, Korhonen I. Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. IEEE Trans Inf Technol Biomed. 2008;12(1):20–6.

Appendix A Image classification: bicycling vs. other

This appendix describes in more detail the algorithm and stages of an image classifier. The classifier is designed to receive data from images, and classify each frame in the category of “walking” or “not walking”. In particular, the classifier is designed to sort data from bicycle activities into the category of “not walking”. The classifier accomplishes this through the use of edge detection algorithms in order to highlight major shapes, which are then consolidated into coordinate data which is sorted through the use of a Naive Bayes statistical algorithm. Ultimately, while the algorithm does demonstrate potential by reaching certain benchmarks of accuracy, more work must be done to ensure and comprehensive and reliable classification structure.

A.1 Introduction

An activity-type classifier required the usage of an image processing algorithm to help an accelerometer sort data points by activity. The image classifier portion was developed to address the specific case of bicycling activities, which otherwise are difficult to distinguish from other activities through accelerometer data alone. After cycling data is filtered as being in the category of “not walking”, remaining frames can then be classified through accelerometer data. The report below describes the implementation of a classifier that is able to filter out bicycling cases by detecting handlebar objects in each frame.

A.2 Design specifications

The classifier must be able to take a frame of camera data from a chest-mounted camera and determine whether the user is riding a bicycle to a reasonable degree of accuracy. The algorithm must run quickly and be able to filter based on a single frame alone.

A.3 Design procedure

The algorithm sought to combine two processes used for identifying and comparing samples of data. The first was the process of edge detection, where sharp changes in an image's data points are located. The second incorporated a Naive Bayes classification algorithm, used to classify test samples of data based on pre-trained categories. In order to format data to be read and sorted efficiently by the classifier, an intermediate stage of data consolidation was used in order to summarize the edge detection results. Java code for the algorithms is presented in 0.

A.3.1 Edge detection

Edge detection was handled through the use of two-dimensional image convolution using smoothing, enhancement, and edge detection kernels. Each of these steps required treating the image as a two-dimensional signal that has a convolution performed on it with a specifically designed matrix, referred to as a kernel. The process of convolution works through multiplying the kernal matrix over the original image grid at various points, with the results being a new image that has been processed to achieve a certain transformation. Three kernels were used for the purpose of this algorithm. The first was a 9 x 9 matrix with the value of (1/81) in each square. This was used for the purpose of blurring the image and thereby smoothing it for the purpose of distinguishing edges. The second kernel involved a 3 x 3 matrix in the form of

```
{-0.3f, -0.3f, -0.3f,
 -0.3f,  3.6f, -0.3f,
 -0.3f, -0.3f, -0.3f}
```

This was used for enhancing certain features of the image to make edges near the foreground more prominent. The third and final kernel was another 3 x 3 matrix in the form of

```
{-1, -1, -1,
 -1,  8, -1,
 -1, -1, -1}
```

used as a bandpass filter highlight areas of contrast and change in the image. The purpose of this stage is to remove background "noise" points that might otherwise be inadvertently detected by later stages of the processor.

The purpose of edge detection in the image classifier was to detect common objects in bicycle activity frames. Because most bicycles have a large handlebar that should take up the center of the camera's perspective, a common set of edges present in bicycle activity data will be a set of prominent horizontal lines in the center. An example of this is displayed in Figure A-1. While the edge detection algorithm was not perfect in eliminating background edges, it does show a general format of output that bicycle activity frames should produce.

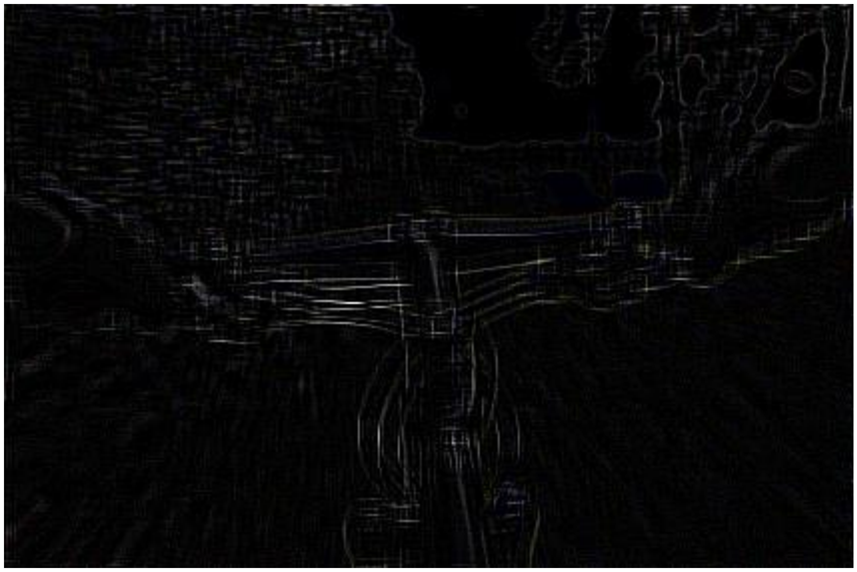

Figure A-1: Image of bicycling with edge enhancement

A.3.2 Data consolidation

After being finalized in an image showing the edges from the first stage of the algorithm, the data was then compiled into two columns to describe the shape data. In this stage, the image was first converted into a two-dimensional array of integers, where the value of each pixel was based on the intensity at that point. A threshold filter was then iterated through the image to capture only points above a certain intensity. The output of this step produced a two-dimensional Boolean array marked as true for each point that satisfied the threshold.

Two ArrayLists are then built based on the number of highlighted edge points in the Boolean array. Each is a list of the edges, with one being a list of x coordinates and the other being a list of y coordinates. Together, they represent the two data columns for the overall shape data.

A.3.3 Classification

The two output data columns are then passed into a Naive Bayes classifier, which is also given training images that describe the corresponding column data for a bicycle activity frame. The classifier then applies the Gaussian Naive Bayes statistical algorithm to determine whether the test data columns share enough similarities with the training data. If they do, the data point represented by the two test columns is classified as similar enough to bicycle activity data, and returned as “not walking”.

A.4 System description

The system makes use of the BufferedImage class to represent image data in a form that can be processed and converted into a basic matrix format. In addition, the ConvolveFilter object, an extension of the AbstractBufferedImage class, is used for the purpose of applying convolution with edge detection and filtering kernels.

The system can construct an edge detection file through the use of the EdgeDetection constructor with a string representing the file path of the test file passed in. The image can then be accessed through the use of the public method EdgeDetection.getImage(), which will return the edge detection image in the form of a BufferedImage object. The edge detection image will be a completely black frame with only the contrast edges in the original test image highlighted.

Next, the ImageProcessor constructor can take in this BufferedImage to produce a series of coordinate data columns. These columns can be accessed using ImageProcessor.getX() and ImageProcessor.getY(). These functions return ArrayLists of x and y coordinate data, respectively, for each highlighted point in the edge detection image.

At the highest end, the ImageClassifier script runs the ImageProcessor and EdgeDetection constructors through a series of images stored in the same file directory. The resulting data obtained from the ImageProcessor methods is stored in a CSV file titled “test.csv”.

A.5 Test plan

In order to validate the algorithm, successful sorting of test data must occur within a satisfactory threshold. The algorithm must be able to accept data from a chest-mounted camera and accurately describe whether or not the user was engaging in the “walking” activity at the time.

A.6 Test specification

Data fed into the system from pictures clearly containing the handles of a bicycle towards the center of the frame should be regarded as definitively in the area of “not walking”. Images that do not fit this description should be sorted in the category of “walking”, as there is not sufficient information to claim otherwise from the image data alone.

In order to ensure a reasonable threshold of accuracy, the system should be able to categorize images properly more than 60% of the time.

A.7 Test cases

A set of 40 training images must first be selected in order to properly calibrate the Naive Bayes classifier. 20 of these will be of bicycle activity and calibrated to “Not Walking”, and will be selected from a set of images with a variety of backgrounds in order to focus the pattern. The remaining 20 will be randomly selected from walking data images. These images will each be run through the edge detection and data consolidation stages in order to be reduced to their x and y data columns. After this, 10 test images will be fed through the edge detection and data consolidation stages, and their x and y coordinate data will be processed by the Naive Bayes classifier.

After these images were chosen from a sample of walking and bicycle data, they were inserted into this process and classified. The result of classification yielded an accuracy of 62%. While this result did achieve test specifications, it does suggest potential weaknesses in the system reliability. From the raw consolidated data formatted (see Appendix C), it is apparent that a distinction exists between cycling and walking activities that can be observed through image processing, though the extent to which this can be observed through statistical classification methods requires further research.

Table A-1 shows a sample of 25 runs of the image classifier. Each image was separated into to 4 horizontal zones, with the count of edge points detected in each zone. Note the bottom of the image frame had clearly different counts of edge points for the two activities. These differences were exploited by the Naïve Bayes classifier.

Table A-1: Count of edge points detected in bicycling and walking

	cycling				walking			
case	top	top-mid	bottom-mid	bottom	top	top-mid	bottom-mid	bottom
1	276	219	174	127	126	88	63	10
2	290	237	181	131	116	91	73	10
3	233	193	152	107	95	85	68	10
4	277	224	151	103	120	75	70	10
5	236	184	142	89	106	68	33	10
6	300	245	166	111	97	67	30	10
7	268	213	153	98	116	79	38	10

	cycling				walking			
case	top	top-mid	bottom-mid	bottom	top	top-mid	bottom-mid	bottom
8	262	214	156	110	164	115	92	10
9	242	205	169	108	152	102	80	10
10	291	250	186	116	156	98	54	10
11	265	225	184	127	172	138	77	10
12	300	261	196	143	146	112	73	10
13	235	206	166	124	165	118	80	10
14	266	230	173	119	203	152	98	10
15	267	226	181	121	202	145	102	10
16	252	202	156	113	186	116	71	10
17	269	214	170	111	178	112	59	10
18	279	241	176	112	189	119	71	10
19	260	218	150	110	190	160	96	10
20	277	224	170	130	207	143	77	10
21	267	209	159	113	213	166	97	10
22	254	213	178	135	201	151	93	10
23	265	221	172	131	219	156	90	10
24	266	206	164	118	163	124	65	10
25	247	196	135	97	199	154	87	10

A.8 Conclusion

The image data that was processed, consolidated, and classified was able to be recognized at a level acceptable by the pre-defined threshold. At 62% accuracy, however, further work is required in order to optimize and ensure reliability of the classifier. It should be noted that frames selected for classification were taken from only three separate videos, and as a result the sample of cycling image frames may not be sufficiently diverse to represent a generalized set of test data. Furthermore, the ratio of training to test data was high, at 80%. Following research should compare the classifier's ability to work from relatively small quantities of training data versus large sample sizes, along with a more diverse set of training and test data.

A possibility considered based on empirical observation of the processing suggests that a simpler classification scheme than Naive Bayes may be viable. The fourth column of data, shown in Appendix C, is significantly different between walking and cycling frames. Applying a threshold of 70 to decide classification would yield a 100% accuracy for this sample of data. Further research should explore the variance in the fourth data column among walking data samples.

Appendix B Java code for bicycling image classification

B.1 Edge detection algorithm

This code presents the operationalized method for edge detection using an input image.

```
package com.jhlabs.image;

import javax.imageio.ImageIO;

import java.util.*;
import java.awt.*;
import java.awt.geom.*;
import java.awt.Image;
import java.awt.image.Kernel;
import java.awt.image.BufferedImage;
import java.io.*;
import java.io.FileInputStream;

public class EdgeDetection {

//   public static void main(String[] args) {
//      BufferedImage sourceImage = load("mountain-bike-handlebars.jpg");
//      BufferedImage destImage = blur(sourceImage);
//      BufferedImage destImage2 = enhance(destImage);
//      BufferedImage edge = edge(destImage2);
//      save(destImage, "blurred.jpg");
//      save(destImage2, "enhanced.jpg");
//      save(edge, "edge.jpg");
   public final static String filePath = "C:\\Users\\nik515\\Documents\\com\\jhlabs\\image\\";

   private BufferedImage edgeDetection;

   //Constructs new EdgeDetection process from an original image
   public EdgeDetection(BufferedImage input) {
      BufferedImage destImage = blur(input);
      BufferedImage destImage2 = enhance(destImage);
      edgeDetection = edge(destImage2);
   }

   //Constructs new EdgeDetection process from an original file
   public EdgeDetection(String file) {
      BufferedImage sourceImage = load(file);
      BufferedImage destImage = blur(sourceImage);
      BufferedImage destImage2 = enhance(destImage);
      edgeDetection = edge(destImage2);
   }

   //Return edge detection image in BufferedImage form
   public BufferedImage getImage() {
      return edgeDetection;
   }

   //Return edge detection image in array form (not used)
   public int[] getArray() {
      int height = edgeDetection.getHeight();
      int width = edgeDetection.getWidth();
```

```
   return edgeDetection.getRGB(0, height/4, width, 3*height/4, null, 0, 0);
}

//Load image from desired filepath
public static BufferedImage load(String name) {
   File source = new File(filePath + name);
   FileInputStream fis = null;
   try {
          fis = new FileInputStream(source);
   } catch(FileNotFoundException fnfe) {
         System.out.println(fnfe.getMessage());
   }

   BufferedImage sourceImage = null;
   try {
       sourceImage = ImageIO.read(fis);
   } catch (IOException e) {
   }
   return sourceImage;
}

//Blurs the image
private static BufferedImage blur(BufferedImage sourceImage) {

   //Define the Kernal to convolve with image
   float[] matrix = new float[81];
        for (int i = 0; i < 81; i++) {
               matrix[i] = 1.0f/81.0f;
   }
   AbstractBufferedImageOp op = new ConvolveFilter( new Kernel(9, 9, matrix));
   BufferedImage destImage = null;
   if (sourceImage != null) {
          destImage = op.filter(sourceImage, destImage);
   }
   return destImage;
}

//Enhances the image
private static BufferedImage enhance(BufferedImage destImage) {
   float[] matrix2 = { 0, -1, 0,
                  -1,  5, -1,
                  0, -1, 0};
   AbstractBufferedImageOp op2 = new ConvolveFilter( new Kernel(3, 3, matrix2));
   BufferedImage destImage2 = null;
   if (destImage != null) {
          destImage2 = op2.filter(destImage, destImage2);
   }
   return destImage2;
}

//Detects edges in the image
private static BufferedImage edge(BufferedImage image) {
   float[] matrix2 = { -1, -1, -1,
                  -1,  8, -1,
                  -1, -1, -1};
   AbstractBufferedImageOp op2 = new ConvolveFilter( new Kernel(3, 3, matrix2));
   BufferedImage destImage3 = null;
   if (image != null) {
          destImage3 = op2.filter(image, destImage3);
   }
   return destImage3;
}
```

```
    //Saves image to desired filepath
    public static void save(BufferedImage image, String fileName) {
        if (image != null) {
            try {
                // retrieve image
                File outputfile = new File(filePath + fileName);
                ImageIO.write(image, "jpg", outputfile);
            } catch (IOException e) {
            }
        }
    }
}
```

B.2 Image classification

This code completes processing of images previously filtered for edge detection.

```
import java.awt.image.BufferedImage;
import java.util.*;

public class ImageProcessor {
    public static final int DIFFERENCE_THRESHOLD = 150;
    public static final int FILTER = 5;
    public static final int CLASSIFIER_THRESHOLD = 100000;

    private boolean[][] image;
    private ArrayList<Integer> x;
    private ArrayList<Integer> y;


    //Constructor that produces boolean array for edge data, along with coordinate columns
    public ImageProcessor(BufferedImage bi) {
        this.image = step1(getImage(bi);
        this.x = solveX(image);
        this.y = solveY(image);
    }

    //Converts BufferedImage into three 2 two-dimensional integer arrays, one for each color
    private static int[][][] getImage(BufferedImage bi) {
        int[][][] pixels = new int[bi.getHeight()][bi.getWidth()][3];
        for (int i = 0; i < bi.getHeight(); i++) {
            for (int j = 0; j < bi.getWidth(); j++) {
                int rgb = bi.getRGB(i, j);
                //red
                pixels[i][j][0] = (rgb >> 16) & 0x000000FF;
                //green
                pixels[i][j][1] = (rgb >>8 ) & 0x000000FF;
                //blue
                pixels[i][j][2] = rgb & 0x000000FF;
            }
        }
        return pixels;
    }

    //Adds three integer values of each point together, then checks whether the result is above a
threshold
    //If it is, the point is added to the boolean array
    private static boolean[][] step1(int[][][] frame1) {
```

```
      boolean[][] rawChange = new boolean[frame1.length][frame1[0].length];
      for (int i = 0; i < frame1[0].length; i++) {
         for (int j = 0; j < frame1.length; j++) {
            int rawDifference = (frame1[i][j][0] + frame1[i][j][1] + frame1[i][j][2]);
            rawChange[i][j] = (rawDifference > DIFFERENCE_THRESHOLD);
         }
      }
      return rawChange;
   }

   //Filters noise by checking if each "true" point has a certain threshold of surrounding "true" 
points
   //Not used in final algorithm, though potentially could be incorporated into further work
   private static boolean[][] step2(boolean[][] prev) {
      boolean[][] filteredDifference = new boolean[prev.length][prev[0].length];
      int n = (FILTER - 1)/2;
      for (int i = n; i < prev[0].length - n; i++) {
         for (int j = n; j < prev.length - n; j++) {
            int marked = 0;
            for (int x = i - n; x < i + n; x++) {
               for (int y = j - n; y < j + n; j++) {
                  if (prev[x][y]) {
                     marked++;
                  }
               }
            }
            if (marked >= FILTER) {
               for (int x = i = n; x < i + n; x++) {
                  for (int y = j - n; y < j + n; j++) {
                     filteredDifference[x][y] = prev[x][y];
                  }
               }
            }
         }
      }
      return filteredDifference;
   }

   //Produces list of x coordinates for each edge point
   private static ArrayList<Integer> solveX(boolean[][] last) {
      int count = 0;
      ArrayList<Integer> x = new ArrayList<Integer>();
      for (int i = 0; i < last.length; i++) {
         for (int j = 0; j < last[0].length; j++) {
            if (last[i][j]) {
               x.add(i);
            }
         }
      }
      return x;
   }

   //Produces list of y coordinates for each edge point
   private static ArrayList<Integer> solveY(boolean[][] last) {
      int count = 0;
      ArrayList<Integer> y = new ArrayList<Integer>();
      for (int i = 0; i < last.length; i++) {
         for (int j = 0; j < last[0].length; j++) {
            if (last[i][j]) {
               y.add(j);
            }
         }
```

```
        }
        return y;
    }

    //Returns x coordinate list
    public ArrayList<Integer> getX() {
        return this.x;
    }

    //Returns y coordinate list
    public ArrayList<Integer> getY() {
        return this.y;
    }
}
```