

**GEORGIA DOT RESEARCH PROJECT 13-13
FINAL REPORT**

**Development of a Portable Bicycle/Pedestrian
Monitoring System for Safety Enhancement**



**OFFICE OF RESEARCH
RESEARCH & DEVELOPMENT BRANCH**

Final Report

DEVELOPMENT OF A PORTABLE BICYCLE/PEDESTRIAN MONITORING SYSTEM FOR SAFETY
ENHANCEMENT

By
Colin Usher
Research Scientist II

Georgia Tech Research Institute

Contract with

Georgia Department of Transportation

In cooperation with

U.S. Department of Transportation
Federal Highway Administration

December, 2016

The contents of this report reflect the views of the author(s) who is (are) responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the Georgia Department of Transportation or the Federal Highway Administration. This report does not constitute a standard, specification, or regulation.

1. Report No.: FHWA-GA-17-1313		2. Government Accession No.:		3. Recipient's Catalog No.:	
4. Title and Subtitle: Development of a Portable Bicycle/Pedestrian Monitoring System for Safety Enhancement			5. Report Date: February 2017		
7. Author(s): Colin Usher, Wayne Daley			6. Performing Organization Code: D7011.0.0.0.0		
9. Performing Organization Name and Address: Georgia Tech Research Institute 640 Strong Street Atlanta, GA 30318			8. Performing Organ. Report No.: 13-13		
12. Sponsoring Agency Name and Address: Georgia Department of Transportation Office of Research 15 Kennedy Drive Forest Park, GA 30297-2534			10. Work Unit No.: N/A		
			11. Contract or Grant No.: PI# 0011821		
15. Supplementary Notes: Prepared in cooperation with the U.S. Department of Transportation, Federal Highway Administration.			13. Type of Report and Period Covered: Final; June 2013-March 2017		
			14. Sponsoring Agency Code:		
16. Abstract: The objective of this project was to develop a portable automated system to collect continuous video data on pedestrian and cyclist behavior at midblock locations throughout the metro Atlanta area. The system analyzes the collected video data and automatically identifies and characterizes the number of pedestrians and their behavior at midblock locations (specifically, their crossing locations and frequency). The system is designed to be portable to support data collection in multiple locations. Some of the pertinent conclusions from this study are:					
<ol style="list-style-type: none"> 1) A portable trailer video recording system was implemented 2) Field testing of pedestrian detection algorithms yielded accuracy of pedestrian detection between 85% and 90% 3) Final output from system tests resulting in pedestrian counts with up to 97% accuracy 					
17. Key Words: Pedestrian, Automated, Tracking, Planning			18. Distribution Statement:		
19. Security Classification (of this report): Unclassified		20. Security Classification (of this page): Unclassified		21. Number of Pages: 87	22. Price:

Table of Contents

List of Tables	v
List of Figures.....	vi
Executive Summary	1
Introduction.....	3
Objectives/Tasks.....	4
1) Work with GDOT to finalize system specifications and test sites	5
2) Complete system design	6
3) Conduct a design review with GDOT	10
4) Purchase/fabricate and integrate system	10
5) Test the prototype system at GTRI facilities	13
6) Develop algorithms to autonomously track pedestrians and cyclists.....	14
Model-based detection	15
Feature-based detection	15
Classification.....	17
Training.....	17
Part-based pedestrian detection.....	18
Conclusion	18
Commercial software evaluation	18
7) Field test the prototype unit and evaluate system operation	32

8) Evaluate the accuracy of the detection algorithms.....	35
9) Conduct training of GDOT personnel	47
10) Data analysis per GDOT request at Clairemont Road	48
11) Develop additional tools for analysis	48
Conclusions and Recommendations	50
REFERENCES	53
APPENDIX 1 – Software Manuals	57
Pedestrian Video Recording Tool Manual	57
Pedestrian Tracking Tool Manual	62
Report Generation Tool Manual	69
APPENDIX 2 – Clairemont Road Analysis Report.....	75

List of Tables

Table 1- Multiple Frame Processing Time Results.....	31
Table 2 - Accuracy Results for First Field Test, October 2014	39
Table 3 - Initial Pedestrian Detection Accuracy.....	39
Table 4 - Maximum Distance of Pedestrian Detection Capability for Field Test Sites	43
Table 5 - Final Accuracy of Pedestrian Detection Algorithms	44
Table 6 - Final Tracker Output Accuracy	45

List of Figures

Figure 1 - Images of Test Site #1.....	5
Figure 2 - Test Site #2	6
Figure 3 - Skycop Cadet Mobile Surveillance System	7
Figure 4 - Netvision Solar Mobile Surveillance Trailer	10
Figure 5 - EMX International Mobile Surveillance Trailer	12
Figure 6 - Pedestrian Detection Output for OpenCV (left) and SLR Engineering (right)	19
Figure 7 - SLR Engineering Pedestrian Trajectory Tracking Image	20
Figure 8 - Techwood Drive Pedestrian Tracking Preliminary Results	21
Figure 9 - Tracker Pre-Processing Steps	21
Figure 10 - Initial Pedestrian Trajectory Output from Techwood Drive.....	22
Figure 11 - Custom Tracker Software Block Diagram.....	22
Figure 12 - Database Table Design	24
Figure 13 - Sample Database View	25
Figure 14 - Image Processing Software User Interface	26
Figure 15 - Mask Generation Dialog.....	26
Figure 16 - Report Generation Tool Dialog.....	27
Figure 17 - Database Logging Test Data	28
Figure 18 - Image Processing Software User Interface	28
Figure 19 - File Structure Format.....	29
Figure 20 - Pre (left) and Post (right) Fixed Trajectory Location	30
Figure 21 - Pedestrian Tracking Software Output.....	33
Figure 22 - Sample Trajectories from Cars Pre (Top) and Post (Bottom) Filtered.....	33
Figure 23 - Tracker Output for 5-Hour Test @ Buford Highway Site #2.....	34

Figure 24 - Pedestrian Tracking Software Output Sample	36
Figure 25 - Vehicle Tracking Software Output Sample.....	37
Figure 26 - Sample Analysis Results	38
Figure 27 - Sample Pedestrian Detector Output	40
Figure 28 - Sample of Pedestrian Double Counted due to Vehicle Crossing.....	41
Figure 29 - Sample of Pedestrian Double Counted due to Tracker Dropout	41
Figure 30 - Multiple Counted Pedestrian Fix.....	42
Figure 31 - Illustration of Pedestrian Size in Pixels.....	43
Figure 32 - Test 2 Visualization Output	46
Figure 33 - Test 3 Visualization Output	46
Figure 34 - Test 4 Visualization Output	47
Figure 35 - Sample Image of a Single Pedestrian Crossing.....	50
Figure 36 - Sample Image of a Single Pedestrian Counted Multiple Times.....	50

Executive Summary

Researchers at the Georgia Tech Research Institute (GTRI) developed and tested a portable pedestrian detection system for characterizing pedestrian behavior at midblock locations throughout the state of Georgia. The system consists of a trailer with four high resolution pan-tilt cameras and a suite of software for collecting video data, processing the video to detect pedestrians and track trajectories, and generate reports based on the tracked results.

Three software tools were developed. A program to enable video recording and saving, a video analysis tool allowing for preparation of video for processing and generating pedestrian trajectory data that is logged to a local database, and a reporting tool that allows for generation of reports and analysis of pedestrian crossing behavior for particular time periods. Results from the report generation tool include a visualization of the pedestrian crossings for the user-defined time period to allow for analysis.

Four field tests were carried out at two locations along Buford Highway in North Georgia. Results from this testing indicate that the system has an average pedestrian detection accuracy of 90%. False detections after filtering of data accounted for approximately 5% to 6%. Final output from the system (including false detections) yielded a pedestrian count that was 96% accurate. Accuracy is particularly high due to the fact that the system only needs to identify pedestrians that are crossing the roads perpendicular to the flow of traffic.

Additional outputs from the reporting tool were developed to assist in analysis of the results from the automatic pedestrian detection algorithms. A comma delimited file for Microsoft Excel records the times and pedestrian identifiers to a file, allowing for plotting and analysis of time-frequency of crossings. In addition, a single image containing all trajectories from a particular

time period overlaid over a snapshot from the video feed allows for visual verification of the tracked trajectories in addition to allowing for determination of density of crossing regions at the location of interest. Finally, images of each individual pedestrian crossing trajectory are generated to allow for verification of the tracking results.

An operational prototype of the system was delivered to the Georgia Department of Transportation (GDOT) at the conclusion of the project. Field testing at additional sites should be executed to allow for a better understanding of system performance under different scenarios and geographic topographies. The current system was designed and tested under the very specific scenario of characterizing midblock crossings. Additional development is required to enable the system to operate and characterize pedestrian behavior at locations other than the midblock.

The research team also identified several areas of potential improvement to add value to the data collected from the system or to allow for further, more advanced analysis of the results. These include recording global position information for crossing data to allow for visualization in a GIS system such as ARC-GIS, providing the ability to view pedestrian crossings in the recorded video streams using the detection database, and adding the ability to separate pedestrians and cyclists by type.

The existing system can also be augmented with the addition of automated vehicle detection, counting, and tracking. This can allow for higher value data and possibly enable more advanced analysis capability. For example, it is anticipated that pedestrian trajectory data coupled with vehicle tracking capability will allow for generation of a “safety” metric. This can be accomplished by analyzing the speed and direction of vehicles coupled with trajectories of pedestrians crossing the roadways.

Introduction

Pedestrians involved in roadway accidents account for nearly 12% of all traffic fatalities and 59,000 injuries each year [1]. Most injuries occur when pedestrians attempt to cross roads, and there have been noted differences in accident rates midblock vs. at intersections [2], [3]. Additionally, vehicle-pedestrian collisions at midblock also tend to be more deadly for pedestrians [2]. This is of significant concern to the Georgia Department of Transportation (GDOT) which is being proactive in exploring various approaches to increasing pedestrian safety [4].

Pedestrian behavior at midblock crossings in the metro Atlanta area is largely unknown. This leads to a lack of information to guide the proper design of lane markings and traffic signals to enhance pedestrian safety. The problem stems from observations that "...When convenient and manageable crossing points are not identified, most pedestrians cross at random, unpredictable locations. In making random crossings, they create confusion and add risk to themselves and drivers..." [5]. Techniques for enhancing pedestrian safety are well known [1], [6]; however, what is lacking are data to support the choice and location of countermeasures at a local level.

The current practice of collecting data on pedestrian behavior is a time-consuming manual process that is prone to error as well as limited in function. Therefore, researchers at the Georgia Tech Research Institute (GTRI) were tasked with the design, development, and testing of an automated system that can be rapidly deployed for data collection to support the analysis of pedestrian behavior at intersections and midblock crossings with and without traffic signals. The collected data can then be used to develop models for input into future intersection and traffic signal/lane marking design. As Atlanta becomes a more pedestrian-friendly city, this

type of data will be crucial to the proper design and development of future crosswalks and midblock signals/markings.

Objectives/Tasks

The objective of this project was to develop a portable automated system to collect continuous video data on pedestrian and cyclist behavior at midblock locations throughout the metro Atlanta area. The system analyzes the collected video data and automatically identifies and characterizes the number of pedestrians and their behavior at midblock locations (specifically, their crossing locations and frequency). The system is designed to be portable to support data collection in multiple locations.

The following tasks were executed by GTRI to achieve the specific objectives:

1. Work with GDOT to finalize system specifications and test site location
2. Complete a system design
3. Conduct a design review with GDOT
4. Purchase/fabricate and integrate system
5. Test the prototype system at GTRI facilities
6. Develop algorithms to autonomously track pedestrians and cyclists
7. Field test the prototype unit and evaluate system operation
8. Evaluate the accuracy of the detection algorithms
9. Conduct training of GDOT personnel
10. Data analysis per GDOT request at Clairemont Road
11. Develop additional tools for analysis

The following sections of this report describe in detail the work completed for each of the tasks listed directly above.

1) Work with GDOT to finalize system specifications and test sites

GTRI researchers met with GDOT personnel and technical monitors Thursday, August 1, 2013, to discuss design criteria of the data collection system. The advantages and disadvantages of a pole mounted- vs. a trailer-mounted system were discussed. On August 19, 2013, GTRI researchers Wayne Daley and Colin Usher met with GDOT representatives Chris Woods and Patrick Werho at the McDonald's located at 3795 Buford Highway NE, Atlanta, GA 30329 (Proposed site #1). The logistics and issues of using a trailer-mounted or a pole-mounted system were discussed. It was decided that a trailer-mounted system would be preferable as long as there is a suitable location that does not impede pedestrian traffic. The system could also be put in the shoulder turn lanes with adequate marking (with cones). Developing a pole-mountable system in addition to the trailer-mounted system was discussed as a future possibility.

The proposed site #1 was visually examined and determined to be well suited for collecting data with a trailer-mounted system. Images of this site are shown in Figure 1. Options for anti-theft and safety measures for the system were also discussed. It was determined that all locations for collecting data must not block pedestrian traffic from the shoulder. The use of both a wheel boot and chaining the system to a fixed object is desired.



Figure 1 - Images of Test Site #1

Site #2 (4050 Buford Highway NE) was also visited to determine the feasibility of using the trailer-mounted design. A suitable location was found, having both adequate space and located in the right-of-way as shown in Figure 2.



Figure 2 - Test Site #2

GTRI researchers met with GDOT personnel Chris Woods and Phil Jackson on Friday, December 6, 2013, to review current pedestrian detection algorithm developments and finalize details on system requirements. The following specifications were determined necessary for successful system operation:

- 1) 500 ft. field of view on either side of system or 800-1,000 ft. on one side of system
- 2) Pedestrian detection accuracy of approximately 90%* or better
- 3) Ability to record number of pedestrian crossings per hour
- 4) Ability to record persistent data for more than 24 hours

*It was discussed that lower than 90% pedestrian detection accuracy could be acceptable due to the system's extended data collection capability.

2) Complete system design

Researchers identified a potential commercially available system from ESI called the SkyCop Cadet Mobile Surveillance System. The system includes a trailer with a hydraulically driven pole,

camera enclosures for three cameras, along with three pan-tilt CCTV cameras. This system includes a set of batteries, a solar panel for charging, a wireless connection to the system via cellular communication, and a DVR for recording video from the installed cameras. The system is shown in the image below and costs approximately \$100,000.



Figure 3 - Skycop Cadet Mobile Surveillance System

While this system contains all the functionality required for a data collection system, the cost is approximately \$30,000 more than initially budgeted for the hardware cost of the system as the manufacturer raised prices from those quoted during the development of the original proposal. The research team negotiated with the supplier and was unable to lower the cost of the system due to the needed requirements.

Several alternatives were identified, and the team compiled price quotes and system capability comparisons to determine which, if any, of the alternatives were acceptable. The systems evaluated were:

- 1) Netvision Solar Mobile Surveillance Trailer with IP Cameras



2) Affordable Video Surveillance Systems Video Security Trailer



3) POGO – Trailer Mounted System from ATD Northwest



The system hardware requirements were finalized. Each of the candidate systems identified were evaluated for their ability to match the requirements. Suppliers of each system were contacted as each system was not capable of doing everything required. All suppliers indicated

they could custom fabricate and add any necessary required features. The hardware requirements list is as follows:

- Trailer has to be of heavy duty welded construction and painted
- 25 ft. extendable camera vertical mast
- Telescoping manual outriggers with adjustable jacks to counter balance full mast extension
- DOT-approved trailer lighting including four marker lights with four-way wire connection
- System must handle wind speeds of up to 60 MPH
- Climate-controlled vent design for cool operation of electronics
- Anti-theft components; removable trailer tongue, dual locking doors, wheel locks, flashing strobe beacon, GPS tracking system, cellular tracking device
- Four 1080P HD cameras PTZ , 20x optical zoom
- Recording System PC: Minimum I7 @3.1 GHz quad core or comparable, 8 GB DDR3 RAM, 4 TB Hard Drive, 10/100/1000MB LAN, USB 3.0
- Wireless Ethernet System
- Cellular card connection capability
- High capacity 600 Watts solar system
- Battery capacity; 900 Amp/Hours with charger
- Hand crank adjustable system for solar charging
- 30 Amp charge controller with PWM charging and LCD diagnostic screen
- 120 VAC shore power
- Electronically controlled automatic switching to provide instant power to equipment
- Main power ON/OFF Switch

- Low and high voltage battery disconnect to prevent under or over discharge of deep cycle batteries

3) Conduct a design review with GDOT

The research team met with GDOT management on March 5, 2014, and conducted a design review. All system alternatives as identified above were presented. The Netvision Solar Mobile Surveillance Trailer (Figure 4) was identified as ideal due to its robust capabilities.



Figure 4 – Netvision Solar Mobile Surveillance Trailer

4) Purchase/fabricate and integrate system

After completing the design review with GDOT, a formal requirements list was generated and the system put out to bid on March 17, 2014. Once purchased, there was an expected lead-time of 4 to 6 weeks for fabrication and delivery of the system.

Five bids were received. The team reviewed each bid in detail. Specifications were found to be lacking as the requirement for the trailer footprint was not specified. This led to lowest bids with systems whose footprint was too large to use on the shoulder of a road. Due to this, a second bid was submitted on April 24, 2014. This bidding process was slated for one week, ending on May 1, 2014.

It was anticipated that the system would be a Netvision system as previously identified.

However, the lowest bidder which met all the requirements was EMX International. An image of the system is shown in Figure 5. The system met all of the following specifications:

- 1) Military grade trailer, 10 ft. length x 9 ft. width when fully deployed
- 2) Nycoil system cable for vertical mast
- 3) Telescoping manual outriggers with adjustable jack-stow on trailer
- 4) DOT-approved trailer lighting including four marker lights
- 5) 60-MPH wind tip over resistance
- 6) Climate-controlled vent design
- 7) Removable/height-adjustable trailer tongue pintel, dual locking doors, wheel locks, flashing strobe beacon, GPS tracking system, cellular tracking device (lowjack)
- 8) Four 1080p HD PTZ cameras, 20x optical zoom
- 9) Dell XPS18 with external 4TB drive for recording
- 10) Wireless Ethernet system
- 11) Cellular card connection
- 12) 560 Watts solar system
- 13) 6 batteries, 1020 Amp/Hours with charger
- 14) Hand-adjustable PV panels
- 15) 30 Amp charge controller
- 16) 120 VAC shore power connection
- 17) Main power ON/OFF switch
- 18) Battery disconnect
- 19) Production facility within 500 miles of Georgia Tech



Figure 5 - EMX International Mobile Surveillance Trailer

EMX International delivered the fabricated system to GTRI facilities during the week of September 15, 2014. The original anticipated delivery date was July 28, 2014. The trailer was delivered 8 weeks behind schedule. The total amount paid to EMX International for the system was \$56,591.

While the system delivered met all the specifications, there was one major oversight related to the trailer design specification. The recording system was designed to record wirelessly to a remote station. Unfortunately, EMX did not anticipate the need for all four feeds to be recorded simultaneously in high definition. The wireless bandwidth was not sufficient for full HD recording of all four cameras simultaneously. Recording from all cameras simultaneously required a physical wired system to the recording PC. In addition, there was no capability to power the recording system directly from the trailer, requiring an outside power source. The EMX engineering team was requested to provide an acceptable solution to resolve the issues. Due to the delay in the initial delivery of the system, the research team requested and was granted a time extension to the project through April 2015.

5) Test the prototype system at GTRI facilities

Initial testing of the data collection system was performed at GTRI facilities. The system was operated outside over the course of two days to test the recording capabilities. Data was recorded of pedestrians crossing the road at several intervals on either side of the system from 50 ft. to 300 ft. This data was recorded using several different data formats and at differing quality settings. Image quality and resolution proved to be sufficient for the required pedestrian detection software (described later). In-depth accuracy analysis was not performed on this data as the team chose to focus on collecting data from the actual sites.

Wireless data collection of the video data proved to be problematic. There was not enough bandwidth available over the wireless connection to stream four simultaneous high definition video feeds. This required direct connection from the recording PC to the cameras via an Ethernet cable. Results from testing showed that direct connection over Ethernet provides sufficient bandwidth for all four HD streams simultaneously.

Hardware testing identified several issues with the power system. It appeared that the solar power charging electronics were faulty which lead to difficulty starting the system up after periods of rest. Typically, if the system was powered off for 24 hours, then it was very finicky to restart. During startup, the solar charger would indicate that there was no charge and fail to provide DC power to the system, even when the system was fully charged. To start the system required multiple iterations of restarting the electronics.

The system was also tested along a stretch of 10th Street alongside Piedmont Park. This location was chosen for the high concentration of cyclists in addition to pedestrians. During this testing, the system's AC inverter failed. This caused power loss to all of the cameras. The system was sent back to EMX International in January 2015 to address the power issues and evaluate why

the AC inverter failed. In addition to repairing the power problems, the system was upgraded to include an on-board recording system. This allowed the system to be deployed without the need for an external generator in addition to enabling complete remote unassisted monitoring. These repairs and upgrades were estimated to cost approximately \$5,000.

The system was repaired and upgraded by EMX International. The total cost for repairs and upgrades was \$5,819.38. Upon receipt, the power system and operation were tested at GTRI facilities. All hardware testing yielded acceptable results. The operating system was upgraded to Windows 8.1 Professional (for an additional cost of \$199) to allow for remote connection and control to the recording PC. Due to the delays caused by the required repairs, the project was granted a no-cost extension through December 5, 2015.

6) Develop algorithms to autonomously track pedestrians and cyclists

At the inception of the project, a comprehensive literature survey was conducted to evaluate the current state of the art and to determine if there are commercial systems available for performing pedestrian detection. Twenty-four papers were identified and reviewed. To summarize, important work done in the field of vision-based pedestrian tracking identified two types of pedestrian detection software approaches [1]:

- 1) Macroscopic – Detecting crowds and masses of people, but not distinguishing individuals
- 2) Microscopic – Detecting individual pedestrians

This project's focus was on the microscopic developments as the ability to detect individual pedestrians for their paths and locations was a required task for tracking their trajectories. Two approaches were used for microscopic-detection methods:

- 1) Model-based detection methods
- 2) Feature-based detection methods

Model-based detection

In model-based detection, an exact pedestrian model is defined and video frames are searched for the matched position as per the model [6]. For matching, Bayesian theory of maximum posteriori probability is generally used. A set of contour examples are used for edge matching and to detect pedestrians. To increase the accuracy of model-based detection methods too many examples are needed which increases matching time. To tackle these issues, some coarse to fine probability matching strategies are used [7]. Markov field models are used to improve robustness of human shape representation [8]. Some 3D model strategies are also developed which attempts to match 3D images taken from stereoscopic cameras in order to improve accuracy [9][10]. General steps to detect pedestrians using model-based methods are background region subtraction and then matching for detection using edges. Texture features are also considered to improve accuracy of detection [11].

Feature-based detection

Feature-based detection is considered the most advanced method for pedestrian detection [6].

The common steps used in this method are:

- 1) Extraction of detection window
- 2) Mapping of image into a feature vector
- 3) Training and classification

Feature selection

Shape features

Edgelet features are the ones which use segments of lines or curves consisting of edges in fixed orientation [12]. Also recent research on Histogram of Oriented Gradient (HOG) by Dalal [13] has received major attention due to the higher accuracy of the algorithm. HOG can describe the local shape and appearance of objects effectively by calculating a gradient value at each pixel. The local and global histogram normalization makes it robust to slight changes in human appearance and lighting conditions. HOG achieved success since it was proposed, and it greatly promoted the development of pedestrian detection. There are many algorithms developed on the basis of HOG using blocks of various sizes [14] to compute HOG, multi-level HOG extracting feature vectors in multi-scale space [15], and co-occurrence HOG - considering the gradient orientation relationship between each pair of pixels [16].

Shape features used with texture and color

This uses a covariance matrix to cover information of position, gray level, and gradient of pixels in a region [17]. Some methods extract a feature set consisting of shape feature (HOG), texture features (co-occurrence matrix), and color features (color frequency); meanwhile, the partial least squares method is applied for dimension reduction [18]. Oliveira makes use of a fusion of HOG features and local receptive field (LRF) features to train classifiers, and the classification results of multiple classifiers are fused [19]. Walk proposes color self-similarity features and combines them with HOG [20]. Wang combines HOG with Local Binary Patterns (LBP) texture features [21]. The HOG-LBP feature is a very discriminative feature even for detecting the head and shoulders of pedestrians. However, it is not wise to combine every feature blindly. On one hand, it may generate features of high dimension which will bring trouble to classification and computation. On the other hand, fusion of some features may blur the discriminative features and cause contrary effects [6]. For example, Wojek performed experiments using fusion of HOG

and Haar-like features, and the detection accuracy was lower than that of only HOG features [22].

Combining with motion features

In this method, features are extracted from frame difference images. Haar-like wavelet features from frame difference images of two consecutive frames are characterized from which both appearance and motion features can be obtained [23]. This is used with multiple consecutive frames for more information. HOG can be computed on optical flow images known as HOF [25]. HOG and HOF are combined to detect pedestrians in low-resolution images [24].

Classification

SVM (Support Vector Machine) is used with linear (for well-discriminating datasets) and nonlinear (for complicated datasets) kernel techniques [13] [15]. Boosting is used to assemble weak classifiers to strong classifiers under the framework of ensemble learning, e.g., adaboost [21]. Walk [26] found that a combination of linear SVM and boosting can effectively decrease the false alarm rate.

Training

Munder [27] did extensive experiments and found that although a perfect combination of features and classifiers can make progress of performance, that progress is much less obvious than increasing the number of training examples. Therefore, enlarging the training set is another way to improve the detection performance. Negative examples which are misclassified in the first round are labeled as hard examples. Those hard examples are mixed with other negative examples to retrain the classifiers [13]. Broggi [28] synthesized pedestrian examples to detect pedestrians in far-infrared images.

Part-based pedestrian detection

This method is used in crowded scenes where it is difficult to detect individual pedestrians [29]. The human body is divided into certain parts, such as head-shoulders, torso, and legs, then the part detectors are trained, respectively. The responses of each part detectors are combined to infer the holistic pedestrians. Part-based pedestrian detection can detect partially occluded pedestrians, but the resolution of images is required to be high enough to make parts of pedestrians explicitly visible. In addition, this inference increases the computation and difficulty [6].

Conclusion

Model-based detection basically defines the exact contours and postures that can be segmented after detection. However, it is hard to define an accurate pedestrian model. In addition, the assumption that the entire foreground is composed of pedestrians is not reasonable in urban outdoor surveillance systems. However, for feature classifier-based methods, pedestrian models are not needed. Instead, specific and various types of features can be used in detection [6]. So for the purpose of this project, it is better suited to use the feature-based approach.

Commercial software evaluation

During the literature review, two software packages were identified as containing pedestrian detection functionality. One such package is the Open Computer Vision Library (OpenCV). This library contains freely available open-source code containing functionality for performing several image processing algorithms. One such algorithm supports pedestrian detection. The second software package identified is a commercially available software package from SLR Engineering, a company located in Austria. They offer a pedestrian detection software package

for system integrators to use in systems, but they do not offer a pedestrian detection system, only the software.

Both software packages were tested. Results showed that both packages perform reasonably at the task of pedestrian detection. While the OpenCV solution contains less false detection, it also contains less pedestrian detection overall. In addition, the SLR software package contains functionality to track pedestrian trajectories which the OpenCV package does not contain. While the OpenCV software is freely available, the SLR package costs approximately \$1,500 per software license. Sample screenshots are shown in Figure 6, with the OpenCV detection in the left-hand side and the SLR detection on the right. The final set of images show detection with trajectories from the SLR software package.



Figure 6 - Pedestrian Detection Output for OpenCV (left) and SLR Engineering (right)

Based on the results from comprehensive testing, the team chose to work with and purchased a license for the SLR pedestrian tracking software library. Testing on publicly available pedestrian video libraries, the OpenCV detector successfully detected 60% of pedestrians with a false

positive detection rate of 2.5%. The SLR tracker detected 80% of pedestrians with a false positive detection rate of 9.5%. In addition, the SLR library contains functionality to track trajectories which is unavailable in the OpenCV library package. A sample image of the SLR pedestrian tracker with trajectories is shown in Figure 7.

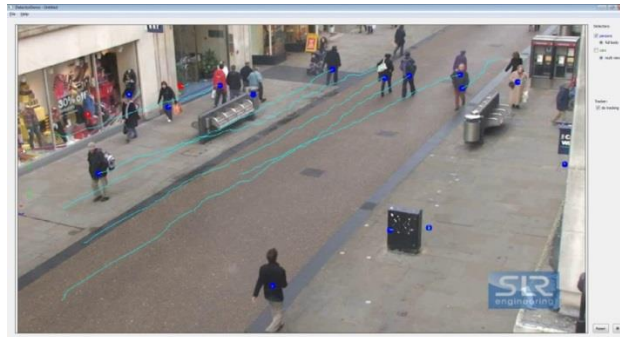


Figure 7 - SLR Engineering Pedestrian Trajectory Tracking Image

The research team began implementation of a custom program using the SLR tracking library to perform pedestrian detection in complex scenes. As anticipated, the tracking algorithms worked well for a very clean set of data (as seen in the image above). Introducing more complex scenes led to several failures of the tracking algorithms. Data was collected on the Georgia Tech campus containing several pedestrians crossing the road on Techwood Drive. This data is more representative of the type of data that will be collected from the detection system. A sample output image is shown below. In this image, there are several false detections contained in the foliage and dark areas of the image.



Figure 8 - Techwood Drive Pedestrian Tracking Preliminary Results

In order to improve both processing time and accuracy, several modifications were made to the image processing algorithms. The detection parameters were adjusted to compensate for the anticipated size of pedestrians relative to the image resolution. In addition, an image mask was generated to remove data from areas in the image where it is not necessary to detect pedestrians. The four images below illustrate a raw input image, the image mask, the resulting image for processing containing the mask, and a sample output image showing pedestrian detection and tracking.

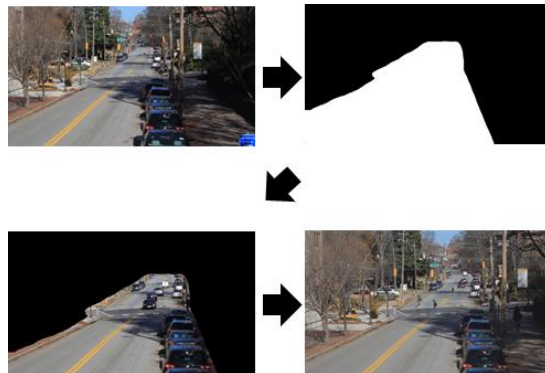


Figure 9 - Tracker Pre-Processing Steps

In order to visualize all trajectories over time, the tracker software was modified to store trajectories in a text file. A Mat-Lab script was developed to parse this data-file and overlay all detected trajectories on an image. This image can be used to visualize density of crossings over a period of time. To test this functionality, a 6-minute video sample containing several

pedestrians crossing the road was processed. The output image is shown below in Figure 10. Pedestrian crossings in the cross-walk are clear to see in the sample image. In addition, several jaywalker crossings are also evident. Software bugs in the tracking algorithm lead to multiple lines drawn for each pedestrian. Modifications of the software were made to eliminate multiple tracks for each pedestrian for more suitable viewing. Figure 10 contains output before this bug was addressed.



Figure 10 - Initial Pedestrian Trajectory Output from Techwood Drive

In addition to the pedestrian tracker development, the team focused on developing both a data handling plan and user experience design. The current custom tracker program flow is detailed in Figure 11. This program loops through video data until no more is available.

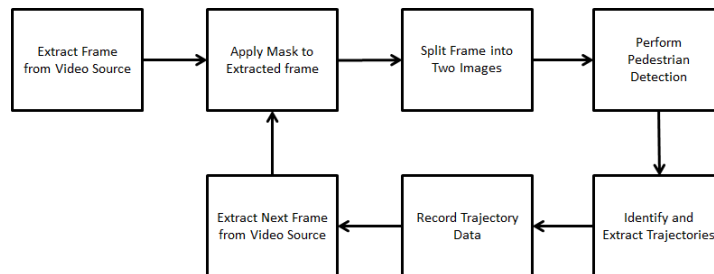


Figure 11 - Custom Tracker Software Block Diagram

The system requires three types of interaction: 1) system setup and data capture, 2) data processing, and 3) data analysis and processing. Each of these requires a different type of interaction. The list below summarizes the user interaction requirements for each:

A) Data Collection Phase

- a. Physical system setup
 - i. Anchor system
 - ii. Extend camera pole
 - iii. Position cameras
 - iv. Initiate the software
- b. Software initiation
 - i. Input site-location identifier (Address, unique ID, etc.)
 - ii. Press 'start' button
 - iii. Verify proper operation (via system status indicator)
- c. Finish data collection
 - i. Access software and press 'stop' button
 - ii. Shut down system
 - iii. Retrieve the trailer

B) Data Processing Phase

- a. Download video data from the trailer system
- b. Access image processing software package
- c. Enter the Site identifier via a text entry box
- d. Select starting video to process from a file-dialog
- e. Press 'start' button

C) Data Retrieval and Visualization Phase

- a. Access the visualization software package
- b. Select the site identifier from a drop-down list
- c. Select the image frame for visualization from a drop-down list
- d. Select the time-span to visualize via text box inputs
- e. Press 'start' button
- f. View the output and select to save or discard

A database is used to store the data regarding the pedestrian detections and their respective characteristics to later use to visualize as required. In this project, the team is using an open-source relational database, MySQL. This database will store information about the site location, and video properties such as date, time, and pedestrian detection coordinates with their respective identifiers. Figure 12 illustrates the database table design.



Figure 12 - Database Table Design

The database design consists of two tables. One will be used to collect trajectory data and the other will contain information about the site location, file directories, and start/end dates. Using a combination of the [Ped_ID], [Site_ID], and [Frame_Number] as the primary key, trajectory information can be uniquely extracted. Figure 13 shows a sample database view, containing all the required information for a single trajectory point.

Site_ID	Site_Desc	S_Date_Time	E_Date_Time	File_Dir_desc
100	North Ave	2014-04-03T14:25:10	2014-04-13T14:25:10	E:/n-ave/...
101	Buford Hwy	004-05-23T14:25:10	004-05-23T14:25:10	E:/b-hwy/...

Ped_ID	Site_ID	Fr_num	X	Y	Height	Width	File
1	101	44	74	106	96	45	E:/n-ave/101.avi

Figure 13 - Sample Database View

Based on the user experience design as described above, Windows dialogs were generated to facilitate the data processing and report generation steps. For brevity, the requirements for the data processing phase are detailed below:

- A) Data Processing Phase
 - a. Download video data from the trailer system
 - b. Access image processing software package
 - c. Enter the Site identifier via a text entry box
 - d. Select starting video to process from a file-dialog
 - e. Press 'start' button

As a part of the image processing software package, the user interface allows for entry of a site identifier (currently a number) and a site description. This identifier and description can be used by the report generation tool described later in this document. The dialog also allows the user to specify which cameras to process (up to four). The initial version of the software only allowed for processing of a single video; future versions added support for four cameras. The initial version of the processing tool is shown in Figure 14.

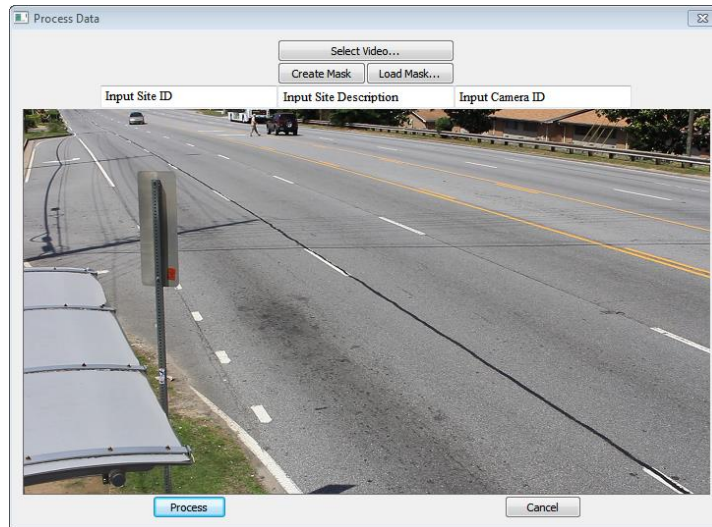


Figure 14 - Image Processing Software User Interface

A secondary dialog allows for a user to create an image mask as described earlier. This dialog loads up an image from the input video, and the user can use mouse clicks to define a polygon which will be masked out (excluded) from the image processing. This allows for faster processing times and rejection of pedestrian trajectories that are not crossing the road. A sample of this dialog is shown in Figure 15. Here, a user can generate a new mask, or load a previously generated mask to apply to the image.

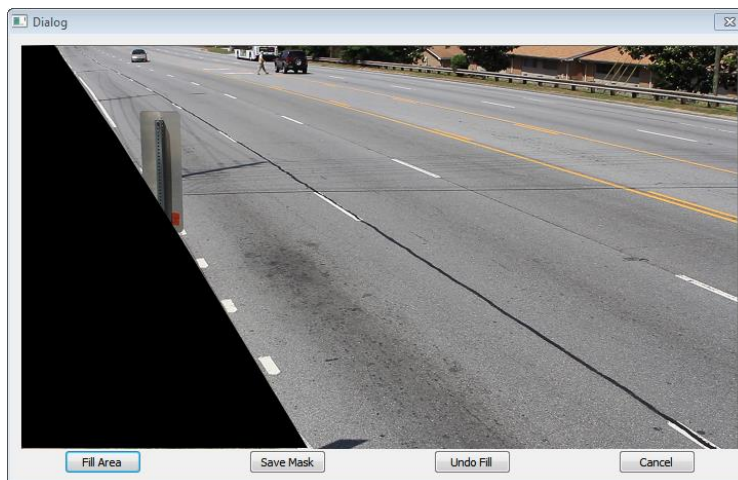


Figure 15 - Mask Generation Dialog

A dialog was also designed to support the report generation. This dialog allows for the user to select a site via the site identifier described above, select a template image with which to superimpose trajectories, and select a time duration within which to process the trajectories. This dialog will also generate the reports in a tabular format for Excel, as well as display the trajectories on the selected images for viewing. The dialog design is shown in Figure 16.

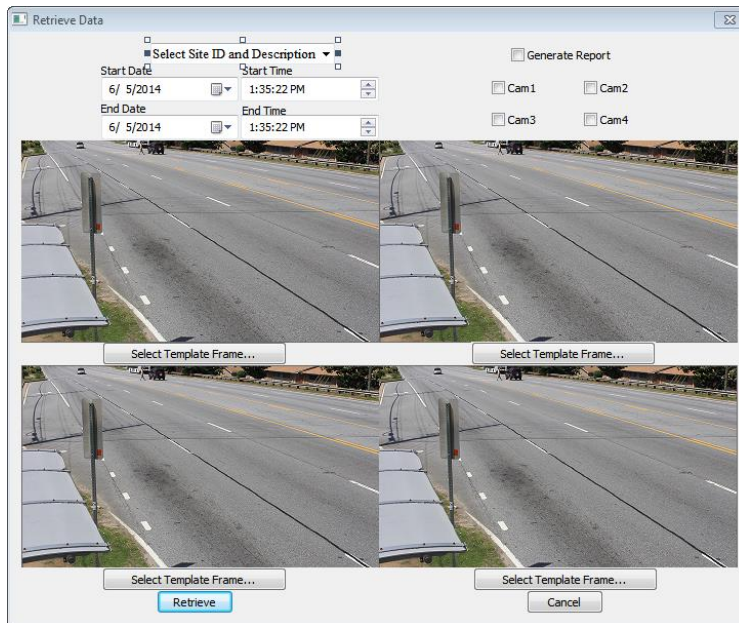


Figure 16 - Report Generation Tool Dialog

Database and logging functionality was tested for collecting data from four cameras simultaneously. To achieve this, the same video was used for each camera input and run through the system. The logged data was then reviewed to look for consistencies. For this test, all four cameras should detect the same pedestrian in the same frame and log accordingly. Figure 17 shows a sample of this data from the analysis. Here, one can see each of the camera detections with the frame identifier and the pedestrian identifier being the same.

frameid	site_ID	Ped_ID	x	y	width	height	topbottom	Cam_ID
17	301	3	1202	43	82	165	whole	1
17	301	3	1199	44	82	165	whole	2
17	301	3	1200	44	82	165	whole	3
17	301	3	1200	41	82	165	whole	4
18	301	3	1184	40	82	165	whole	1
18	301	3	1185	41	82	165	whole	2
18	301	3	1186	41	82	165	whole	3
18	301	3	1183	39	82	165	whole	4
19	301	3	1166	38	82	165	whole	1
19	301	3	1167	39	82	165	whole	2
19	301	3	1173	39	82	165	whole	3
19	301	3	1169	36	82	165	whole	4
20	301	3	1153	37	82	165	whole	1
20	301	3	1154	37	82	165	whole	2
20	301	3	1160	37	82	165	whole	3
20	301	3	1156	34	82	165	whole	4
21	301	3	1140	35	82	165	whole	1
21	301	3	1140	35	82	165	whole	2
21	301	3	1146	35	82	165	whole	3
21	301	3	1138	32	82	165	whole	4
22	301	3	1135	45	62	124	whole	1

Figure 17 - Database Logging Test Data

The first version of the software only allowed processing for one camera. The current software, shown in Figure 18, now supports processing for up to four cameras simultaneously.

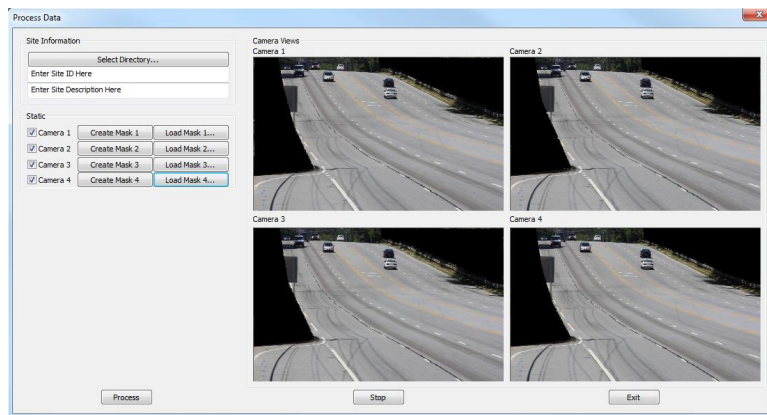


Figure 18 - Image Processing Software User Interface

A file-system structure was designed for storing the video files for processing. This structure allows the program to load and process from the four cameras, while also defining an organized method of storing the files. Figure 19 illustrates this file structure. A folder named 'PedmonData' consists of one subfolder for each data collection site. Each site folder contains a subfolder for each camera, within which the video data and image masks are stored.

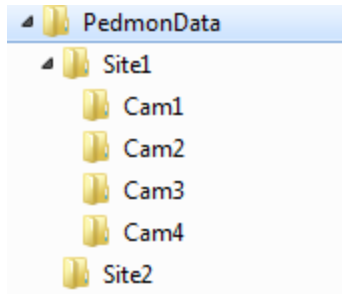


Figure 19 - File Structure Format

A second program was also designed and implemented to allow the tracker to run in a Red-Hat Linux environment. This program is essentially the processing engine run in a command window environment with no user interface. This development allowed for testing of processing on a computing cluster located at GTRI facilities. No testing was performed on the Linux version of the tracker for parallel processing. Time and budget constraints led to the need to eliminate this testing from the project.

With the development of the Windows-based user interface complete, software testing was performed on test data collected both at Georgia Tech facilities and at the proposed data collection site #1 on Buford Highway. Several bugs related to the licensing of the pedestrian detection routines were identified and addressed.

The tracker license only allowed for instantiation of a single tracker, whereas the tracker software uses up to eight simultaneous trackers in order to reduce the processing time using multiple cores on the processor. Attempts to fix this licensing issue with the hardware USB dongle provided from SLR proved unsuccessful. SLR provided the team with a temporary unlicensed version of the tracker to continue development until they solve the licensing issues on their side.

Another software bug was identified related to running multiple trackers simultaneously on the PC system. Running the software system with eight trackers leads to freezing of the tracker

software. This appeared to be related to the SLR libraries. To circumvent this issue, the tracker is only processing two cameras at a time, which utilizes four tracker instances (two trackers per camera).

A bug was identified in the visualization package and subsequently fixed. The previous version of the software would generate trajectories for pedestrians with an offset based on the top of the pedestrian. The intention was to track the location of the pedestrians' feet on the road surface, to reflect the accurate location of crossing. The visualization tool was updated to fix this issue, and tracks are now with respect to the pedestrians' feet location on the pavement surface as shown in Figure 20. The following screenshot shows visualization results before and after the fix. Notice the high number of pedestrians' trajectories in the crosswalk. The left image shows the incorrect trajectory location, whereas the right image shows the corrected visualization.



Figure 20 – Pre (left) and Post (right) Fixed Trajectory Location

Initial versions of the pedestrian tracking algorithm resulted in poor processing time for each image frame. Significant efforts were made throughout the project to improve the processing time. Modifications made specific to the tracker to improve processing time included splitting the image into two sub-images to be processed in parallel on multiple cores. A multi-threaded implementation allows the PC to process each image in a separate core, more than doubling the processing time. Initial processing time per frame was approximately 1,400 milliseconds.

Parallelized methods were able to process each frame at an average of 598 milliseconds, a 230% improvement.

Different approaches to parallel processing of the images were implemented and tested. It was determined that splitting the images was both unnecessary and complicated the processing setup requirements, thus it was removed. Processing time for one frame was further reduced to an average of 215 milliseconds per frame (from a previous average of 598 milliseconds). The team worked closely with SLR Engineering, the software company providing the tracker algorithm, to further improve the processing time.

Time tests were performed for processing video data from multiple cameras simultaneously. This is because the system has four cameras, and simultaneous processing taking advantage of the PC's multiple processing cores could reduce the time required. This test was performed on an Intel Core i7 870 CPU running at 2.93 GHz with 8.0 GB of RAM. Running on a more powerful computer could result in better results. Table 1 shows the results from this testing.

Table 1- Multiple Frame Processing Time Results

Simultaneous Frames Processed	Process Time Per Frame	Total Frames Processed Per Second
1	215	3.92
2	257	7.78
3	305	9.84
4	355	11.27

Scenes with more pedestrians and objects such as cars and buildings negatively impact the processing time. Scenes that are simple process faster, whereas scenes that are more complicated process slower. This is a result of the algorithms attempting to robustly eliminate false positives from the detections.

Further refinement and testing showed that the system could successfully track pedestrians at as low as six frames per second in the video. This was achieved by only processing every fifth frame from each video stream. This results in faster processing times for the system by reducing the number of frames required to be processed. Processing time for four simultaneous video streams at six frames per second (parallelized) results in approximately a 2-to-1 processing time requirement per unit time, meaning that for each hour of video data, it will require approximately 2 hours of processing time. For processing a single video stream, the time requirement is approximately 1.3 to 1. It is anticipated that faster processors can potentially achieve real-time processing of the trajectory data. The user manuals describing functionality of each of the individual programs described in this section are contained in Appendix 1 (A-C).

7) Field test the prototype unit and evaluate system operation

The team took two initial trips to data collection site #1 on Buford Highway in late 2013. Data was collected of both random pedestrian crossings and of team personnel crossing the midblock at varying distances from the collection system. Video was captured using a Canon 60D DSLR camera in 1080p resolution to simulate the expected data collection system resolution. The camera was located on the top of a ladder mounted in a truck bed to achieve proper height and field of view. This data was used to test the tracking algorithms with respect to the data collection sites, and to understand the field of view and camera setup required for the image processing algorithms. Figure 21 shows a sample screenshot containing a tracked pedestrian crossing in the midblock location. The yellow line illustrates the tracked trajectory.



Figure 21 - Pedestrian Tracking Software Output

During this early testing, the detection algorithms were able to detect all of the pedestrian crossings. However, several cars were also falsely detected leading to invalid trajectories. Algorithms were modified in the Report Generation Tool to remove trajectories from cars travelling in the lanes based on the trajectory slope, but cars crossing the road perpendicular to the lanes still resulted in false trajectories. Figure 22 shows a sample image containing trajectories from vehicles (blue lines) and a pedestrian pre- and post-filtered. Generating the slope for removing vehicle trajectories is described in the user manual for the Report Generation Tool located in Appendix 1B.

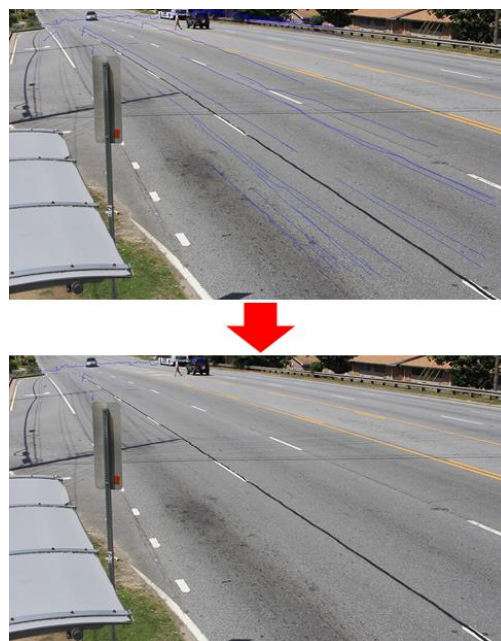


Figure 22 - Sample Trajectories from Cars Pre (Top) and Post (Bottom) Filtered

After obtaining the trailer-mounted recording system in September 2014, it was deployed and data was recorded for approximately 2 hours at site #1 on October 14, 2014. During this time, the team established a protocol for testing the detection algorithms based on results from the previous testing and analysis. This protocol is described in the following section.

The trailer was then deployed to site #2 on Buford Highway, and data was recorded for approximately 5 hours on March 4, 2015. Testing followed the same pattern as testing for site #1 where random pedestrian crossings were recorded in addition to the research team crossing at several intervals on both sides of the system. All accuracy results from this test are presented in the following section. Sample output from the visualization tool for all four cameras is shown in Figure 23.

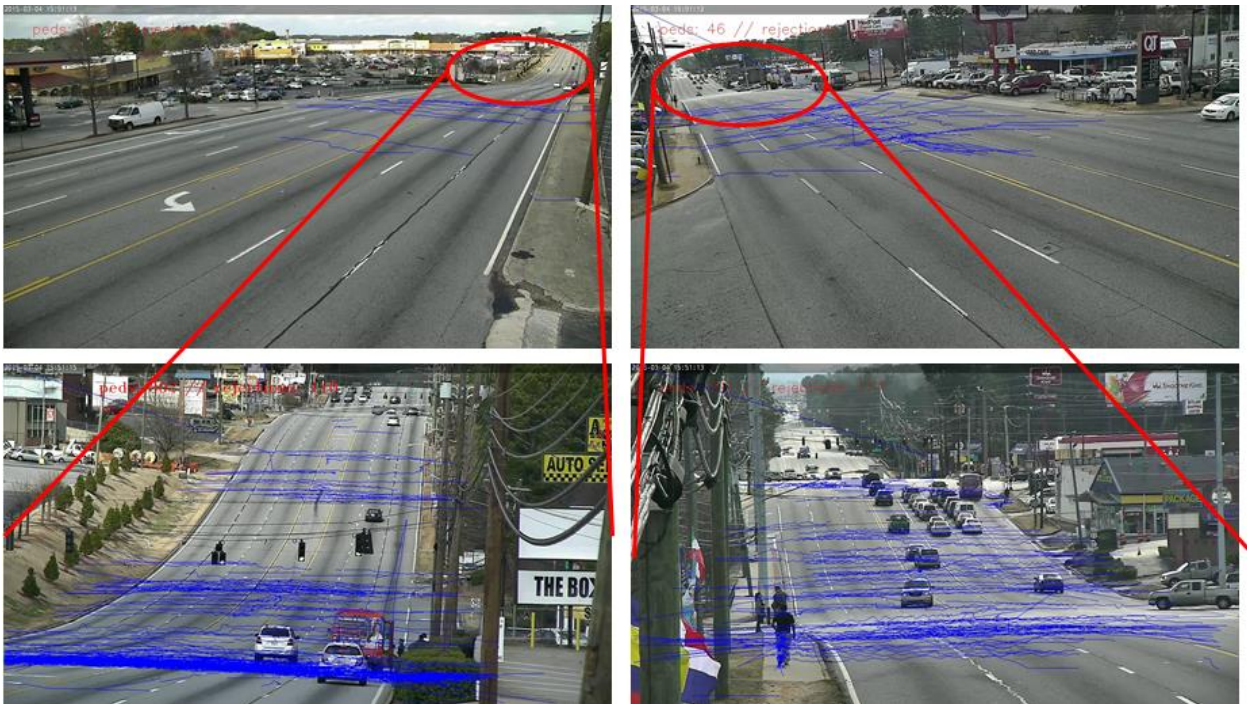


Figure 23 - Tracker Output for 5-Hour Test @ Buford Highway Site #2

In the images above, the red circles indicate the area zoomed into in the bottom images. This is a good illustration of the system's capability to record and track pedestrians both close to and very far from the system. In just this 5-hour time period, approximately 500 pedestrians were tracked.

The system was setup and enabled to run for 5 days at data collection site #1 on Buford Highway on March 20, 2015. However, after 4 days, the system battery was too low to continue recording, so it was removed from the site and returned to GTRI facilities. The team concluded that due to rain and cloud cover for the duration of the test, the system's solar panels were not operating at full efficiency. The addition of the on-board computer also draws more power on the system, reducing its effective up-time. During this time, the system collected both random pedestrians and the research team crossing the road at several intervals in front of the system on both sides.

The system was setup and enabled to run at data collection site #2 on Buford Highway in May 2015. During this collection, the system ran for approximately 6 days and nights. During this time, the system collected both random pedestrians and the research team crossing the road at several intervals in front of the system.

All results related to the pedestrian tracking accuracy for the above-described field tests are presented in the following section.

8) Evaluate the accuracy of the detection algorithms

Accuracy testing was carried out at several times throughout the entire development of the system. This section describes the timeline and results obtained throughout the project, with the most recent comprehensive results presented at the end.

SLR Engineering consistently provided new pedestrian tracker updates throughout the project containing improvements to the tracking routines as well as updated pedestrian classifiers trained on data taken from the data collection sites on Buford Highway. Tracker updates consistently operated significantly better, detecting the pedestrians with less false detection of objects such as vehicles and improving results in shaky video (due to high winds). Figure 24 shows a sample image from the detector with two pedestrians crossing the highway in opposite directions.



Figure 24 - Pedestrian Tracking Software Output Sample

In addition to the tracker updates, SLR provided a demo of their vehicle tracker. Initial observation of the vehicle tracker indicates high accuracy in the vehicle detection capabilities, but a comprehensive test was never carried out. Figure 25 shows a sample output of the vehicle tracker.

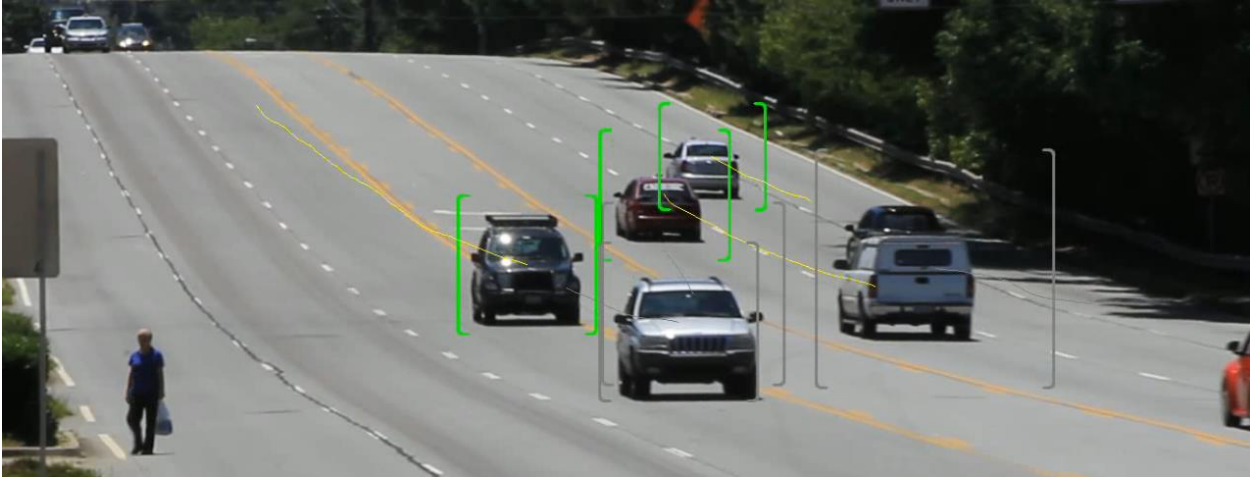


Figure 25 - Vehicle Tracking Software Output Sample

A testing protocol was developed for the field tests. After the system was deployed and configured, team personnel crossed the roadways for the first hour on both sides of the system at approximately 50-foot intervals (where possible) throughout the entire system field of view. This ensured pedestrian crossing data for the accuracy tests to be carried out. During this time, random pedestrian crossings were also captured. The system was then run for multiple days until the batteries required recharging, at which time the system was returned to GTRI facilities. For the video data, random 1-hour time-slices were selected and analyzed for detection accuracy. Special attention was paid to times of detrimental video quality, such as during heavy rain. Several sample images from the software output are shown in Figure 26. The images show successful tracking of pedestrians at several distances from the system. The image at the bottom is the result of the tracker shown for several midblock crossings.



Figure 26 - Sample Analysis Results

Preliminary accuracy testing was performed on the data collected at the Buford Highway site #1 in October 2014. The software algorithms were run on the video collected during the test, and Table 2 summarizes the results. It is important to note that the team expects there to be lower accuracy in each individual camera. This is by design as two cameras are used to track pedestrians on a single side of the system. One camera is zoomed in to collect data for pedestrians far from the system, and the other camera is zoomed out to capture pedestrians near the system as illustrated in Figure 23. The accuracy below is presented for each independent camera and then finally for the combined results. The combined accuracy is the important statistic here, as it indicates when a pedestrian is tracked in either one of the cameras, resulting in a successful track.

Table 2 - Accuracy Results for First Field Test, October 2014

Camera	Number Pedestrians	Detected	Accuracy
1	17	14	82.35%
2	19	10	52.63%
3	16	10	62.5%
4	15	12	80%
Combined	67	59	88.06%

Initial accuracy analysis of the final pedestrian detection capability of the system was completed in April 2016. This includes four field tests, two at each site #1 and #2, all carried out in 2015. For each of these tests, the following method was used to determine accuracy of pedestrian detection:

- 1) In the first hour of operation, GTRI personnel manually crossed the midblock location at approximately 20- to 50-foot intervals for the entirety of the system field of view.
- 2) After acquiring data, all pedestrians visible in the camera images were manually counted (including GTRI personnel and other miscellaneous pedestrians).
- 3) The pedestrian tracker was then run on the video data containing the crossings.
- 4) Properly identified pedestrian accuracy was manually calculated.
- 5) Accuracy is tabulated combining the detection from each camera, two cameras facing the same direction.

The results from the four tests are summarized in Table 3 below.

Table 3 - Initial Pedestrian Detection Accuracy

Test	1	2	3	4
Accuracy (%)	88.06	87.39	85.29	93.5

This initial accuracy testing only took into account if a pedestrian crossed and was properly identified by the system. It did not include false detections on non-pedestrians. A selection of tracked pedestrians from testing is shown in Figure 27.



Figure 27 - Sample Pedestrian Detector Output

During accuracy testing, one serious issue was identified with the pedestrian tracking algorithm.

For several successful detections, the detector would double and triple count a pedestrian due

to short dropouts of the detection. This occurs mostly when a vehicle passes through the field of view and physically blocks the pedestrian for an extended period of time. One sample of this is shown in Figure 28. Occasionally, the pedestrian tracker also loses track of pedestrians for a very short period. This occurs mostly when a pedestrian is at the maximum distance from the system, and appears very small and blurry to the system. A sample of this is shown in Figure 29.



Figure 28 - Sample of Pedestrian Double Counted due to Vehicle Crossing



Figure 29 - Sample of Pedestrian Double Counted due to Tracker Dropout

Working with SLR Engineering, the suppliers of the pedestrian tracking software library, modifications were made to the tracking algorithm related to estimating pedestrian locations based on previous detected movement. The modifications were all contained in the configuration files for the tracker and did not require any rebuild of the software. However, these modifications are not straightforward, and it is expected that a user of the system will have to be familiar with the actual inner-workings of the algorithm in order to understand how to properly affect the processing accuracy. Figure 30 illustrates a before and after detection of a

pedestrian that was counted three times in the left-hand frame, and counted accurately once in the right-hand frame (also illustrating other pedestrian crossings). Of importance is the single connected trajectory through the middle of the frame instead of the three separate segments.



Figure 30 - Multiple Counted Pedestrian Fix

Final Analysis for camera 1 and camera 3 for site #2 was completed in August 2016. Both cameras face north on Buford Highway, covering the near and far views of the system. Initial analysis was conducted using all pedestrians visible to the system in the video data. After discussions with SLR Engineering, it was pointed out to the team that the tracking algorithms accuracy is not defined for pedestrians that are less than 30 pixels tall. To illustrate this point, the images in Figure 31 show pedestrians in camera 1 that are larger (top image) and smaller (bottom image) than 30 pixels tall.

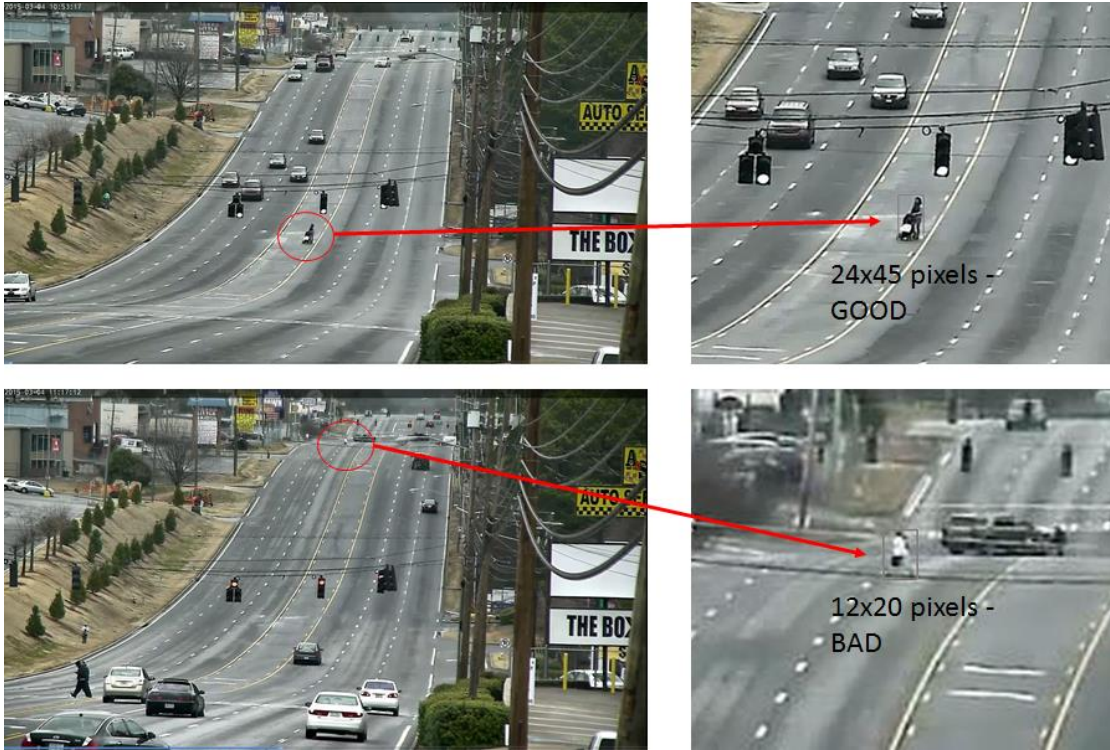


Figure 31 - Illustration of Pedestrian Size in Pixels

Because the initial analysis presented in Table 3 counted all pedestrians, including the ones that were too small for the detector, a second analysis was performed to refine the impact on the accuracy for all field tests. In addition, the maximum pedestrian detection distance was calculated for each site. Table 4 documents the maximum distance of acceptable pedestrian detection from the recording system on each side of the system.

Table 4 - Maximum Distance of Pedestrian Detection Capability for Field Test Sites

Test Site	1	2
Maximum Distance North (feet)	725	1782
Maximum Distance South (feet)	888	1495

Re-visiting the accuracy of the pedestrian detector and taking into account only those pedestrians that were within the pixel specification of the tracking algorithms, the following accuracy was tabulated and is presented in Table 5. Please note that test 1 was not re-visited due to the video file naming convention with the newest version of the software, and therefore,

was unable to be re-processed. Tests 2 through 4 were re-processed, and the accuracy did increase slightly (< 1%) for all cameras. Also note that this testing only identifies the accuracy of the pedestrian detector, meaning that false positives such as vehicles that are errantly tracked are not included in these results. Accuracy of the system with respect to pedestrian trajectories and false trajectory filtering is presented later in this section.

Table 5 - Final Accuracy of Pedestrian Detection Algorithms

Test	1	2	3	4
Accuracy (%)	88.06	87.5	90.9	93.81

The paragraphs directly above describe the accuracy of the pedestrian detection algorithms in a stand-alone fashion. In reality, the end result from the system must generate a total reported count of pedestrians for a particular time-span. The tracker algorithms frequently mistake vehicles as pedestrians due to several factors. Due to this, algorithms were developed to filter likely false trajectories based on their slope. See Section 7 for details on this slope-filtering method.

Table 6 details the final system accuracy including the actual number of pedestrians, the number of pedestrians successfully detected, the number of false detections due to vehicles or single pedestrians counted multiple times, and the final result of the tracker with respect to the actual pedestrian count. Please note that the final result is the reported number of pedestrians as classified by the system, including false detections. This illustrates the expected performance of the system when a user takes the singular result of the system output. Analysis tools are available for visualizing pedestrian crossing data in images and for validating the tracker results. These tools are described in Section 11 titled ‘Develop additional tools for analysis’. Complete tracker output for Test 1 is not presented due to updates in the processing tool software

requiring a video naming format that was not adhered to at the time of the first test video capture. Therefore, test 1 videos were unable to be reprocessed.

Table 6 - Final Tracker Output Accuracy

Test	Actual Pedestrians	Detected Pedestrians	False Detections / Multiple Counts	System Output (Includes false positives)	Final Output Accuracy w.r.t Actual (%)
1	67	59	N/A	N/A	N/A
2	108	91	17	104	96
3	44	42	0	42	95
4	100	91	6	97	97

Please note that the final output accuracy in the right column describes the accuracy of the system output that results from taking the tracker results *as-is* after processing. This number potentially includes false detections from vehicles and multiple counts for single pedestrians (as listed in the false detections column). It is assumed that a user will use this output from the tracker when analyzing the reports. However, there are methods and tools to allow the user to refine the results. These are described in Section 11 below.

After generating a report, the tracker also generates an image with all pedestrian trajectories overlaid on it as described in Section 6 of this report. Figure 32 through Figure 34 contain the visualizations for all cameras for tests 2 through 4 summarized above in Table 6. These visualizations allow for determining where on the road the bulk of pedestrian crossings occur, as well as allowing for sanity checking of the crossing data.

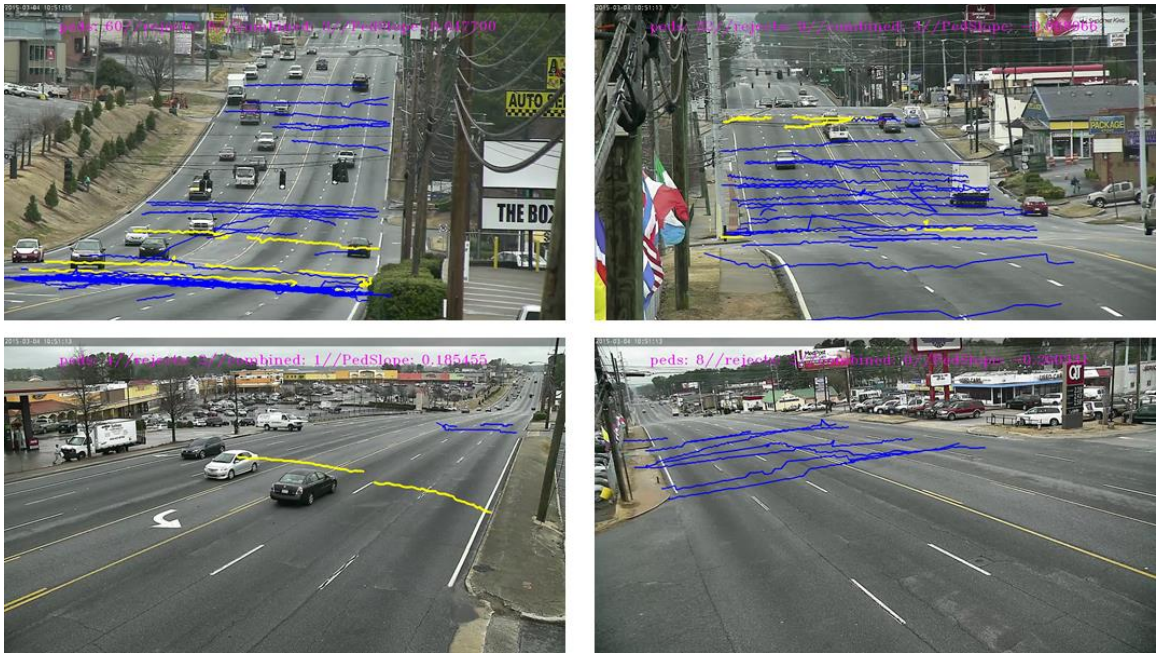


Figure 32 - Test 2 Visualization Output

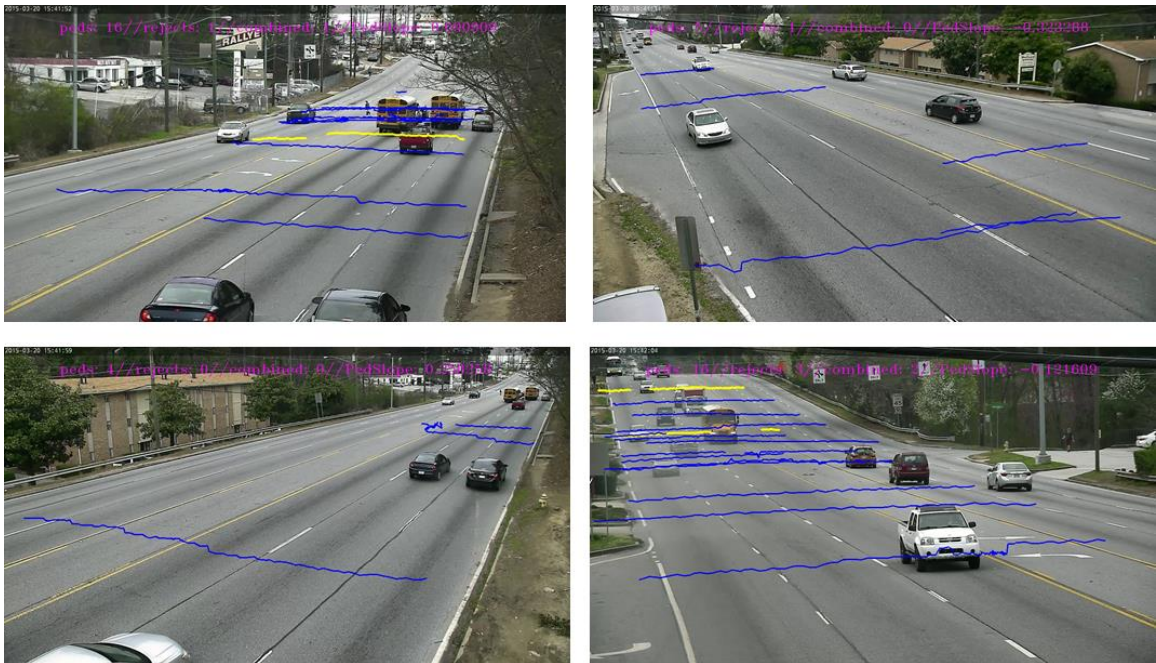


Figure 33 - Test 3 Visualization Output

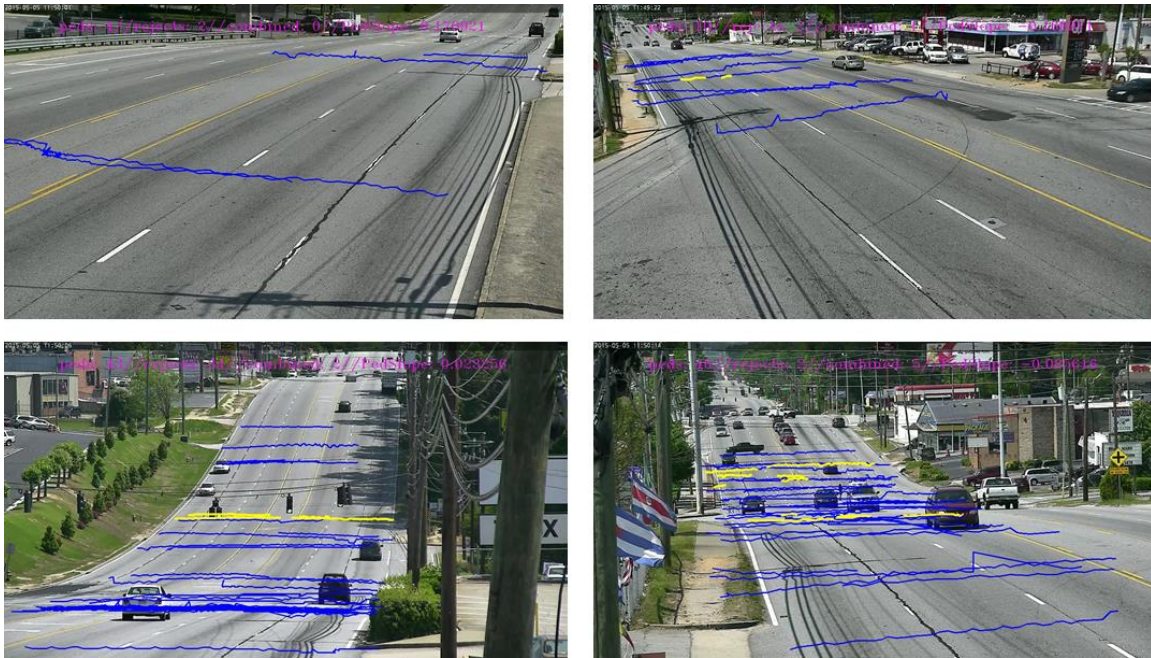


Figure 34 - Test 4 Visualization Output

9) Conduct training of GDOT personnel

After a significant (greater than 6 months) delay due to internal accounting issues, additional project funds and a time extension through December 11, 2016, was granted primarily to facilitate training and demonstration of the system for GDOT personnel. Initial training for using the video recording system and trailer was conducted on July 6, 2016, during a data collection trip as described in the following section. During this trip, the trailer system was presented and the recording system setup and operation was demonstrated.

Final training and demonstration of the system was completed on January 31, 2017. Training was performed at GTRI facilities. The trailer recording system was demonstrated and a hands-on training was performed. This training encompassed the operation of all software tools and deployment and management of video data. Training was attended by the following GDOT personnel:

- Katelyn Digioia – State Bicycle and Pedestrian Engineer
- Sarah Lamonthe – Transportation Specialist 2, Research
- Yogendra Patil – Transportation Specialist 2, Research
- Patrick Weaver – District 3 Traffic Operations Specialist
- Donnie Boyd – District 5 Traffic Operations Manager
- Edlin Regis – District 7 Traffic Operations Manager

10) Data analysis per GDOT request at Clairemont Road

A special request was made to collect data with the system at the intersection of Maedris and Clairemont roads. This analysis was completed, and a report summarizing the results is attached in APPENDIX 2 – Clairemont

Road Analysis Report.

11) Develop additional tools for analysis

In addition to the image file generated at the end of report generation using the Report Generation Tool (as described in Section 9), additional tools for analysis of pedestrian crossing behavior were developed to allow for more flexibility with analysis of the results. This includes generation of a .csv file containing report results and a set of individual images illustrating each pedestrian crossing. These reports are generated in the C:\Pedmon\REPORTS folder. Each report is placed in a subfolder, named pedmonReport-<Site ID> where the <Site ID> is the identifier assigned to the video recordings. See the Pedestrian Tracker Software Manual in Appendix 1 for details on the Site ID.

The .csv file is a comma delimited file designed to be opened in Microsoft Excel. This file contains header information pertaining to the report, such as user settings and an overall summary of the number of pedestrians. The header information is followed by a single row for each detected pedestrian crossing. Each row contains the pedestrian ID number and the time that the crossing occurred with respect to the video time. This allows for further analysis such as time-frequency of crossings. This also allows for presentation of the data as shown in the Clairemont report contained in Appendix 2.

Individual images containing each pedestrian crossing as detected by the system are also saved locally in the associated report folder. This allows for review of the detector results for checking of accuracy. A user can quickly thumb through these images and do a sanity check on the trajectories and manually append the pedestrian count if so desired. Each image contains the pedestrian ID, the time of crossing, and a line indicating the tracked location of crossing. For example, observing these files can help a user determine if the tracker missed combining two trajectories that should belong to a single pedestrian. This can be done by observing the trajectory output and the time of the crossing. Figure 35 illustrates a single pedestrian crossing. Figure 36 shows a case where the tracker should have combined two trajectories into a single one. Here, you can see the close proximity of the trajectories, and the time of detection difference. The image also details the direction of crossing with an arrow.

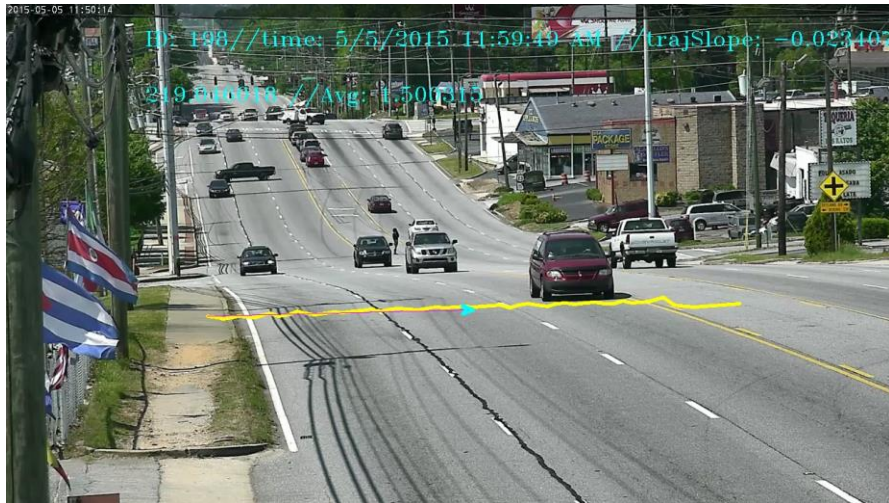


Figure 35 - Sample Image of a Single Pedestrian Crossing



Figure 36 – Sample Image of a Single Pedestrian Counted Multiple Times

Conclusions and Recommendations

A portable video capture system for characterizing pedestrian crossings at midblock locations was developed and tested. A trailer containing four high definition pan-tilt cameras was acquired from EMX International. Software for acquiring video data, processing data for pedestrian detection, and generating reports based on pedestrian trajectories was developed. A database was developed to store the pedestrian detection data resulting from the video processing software.

Four field tests were carried out at two locations along Buford Highway in North Atlanta. For these tests, the pedestrian detection accuracy averaged 90.06%. False detections after filtering

of data accounted for approximately 5% to 6%. Final output from the trailer (including false detections) yielded a pedestrian count that was 96% accurate.

Additional tools were developed to allow users to validate tracker results. These tools allow for visualizing the total pedestrian behavior in a single image for a set period of time, in addition to viewing individual trajectories for each pedestrian. A comma delimited excel file (.csv) is generated with time stamps and pedestrian identification numbers to allow for generating time-sequence plots of pedestrian crossings.

High accuracy is obtained by the system due to the fact that the algorithms are designed to find pedestrians that are crossing perpendicular to the flow of traffic. This allows for easy removal of false positives. If this system is to be used to perform pedestrian tracking in places other than midblock crossings, future work will be needed to develop more robust tracking accuracy for pedestrians that may be travelling in the same direction as the flow of traffic.

There are several opportunities to improve the system's reporting capabilities. These include the ability to determine global positioning data for crossings to allow data to be plotted in a GIS system, providing the ability to view pedestrian crossings in the recorded video streams using the detection database, and adding the ability to separate pedestrians and cyclists by type. The current system identifies both pedestrians and cyclists, but it currently does not characterize them separately.

The image processing tool was designed to characterize pedestrian behavior in any video stream. It would be prudent to test this using data streams from other sources around the city such as CCTV camera feeds. There is an opportunity with the Midtown Alliance to obtain sample data from their Midtown Traffic Operations Program (MTO) work.

Finally, adding vehicle detection capability to the system would allow for augmenting the pedestrian data with vehicle counts and frequencies. The research team is interested in exploring the ability to identify and characterize the behavior and interaction between pedestrians and vehicles. In particular, there is interest in exploring the ability to generate a metric that encapsulates the potential danger for crossings at particular locations with respect to crossings at other locations within a geographic area.

REFERENCES

1. Redmon, Tamara. "Evaluating pedestrian safety countermeasures." *Public Roads* 74.5 (2011).
2. Chu, Xuehao. "Pedestrian safety at midblock locations." (2006).
3. Cui, Zhenzhong, and Shashi Nambisan. "Methodology for evaluating the safety of midblock pedestrian crossings." *Transportation Research Record: Journal of the Transportation Research Board* 1828 (2003): 75-82.
4. M. Pirkle, *State DOT Looking to Reduce Dangers*, Atlanta, GA: Atlanta Journal Constitution, 2013.
5. Turner, S., et al. "FHWA university course on bicycle and pedestrian transportation." *Draft Report, Federal Highway Administration, McLean, VA* (2004).
6. Li, Bo, Qingming Yao, and Kunfeng Wang. "A review on vision-based pedestrian detection in intelligent transportation systems." *Networking, Sensing and Control (ICNSC), 2012 9th IEEE International Conference on*. IEEE, 2012.
7. Gavrilu, Dariu M. "A bayesian, exemplar-based approach to hierarchical shape matching." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.8 (2007): 1408-1421.
8. Wu, Ying, and Ting Yu. "A field model for human detection and tracking." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.5 (2006): 753-765.
9. Zhao, Tao, and Ramakant Nevatia. "Bayesian human segmentation in crowded situations." *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*. Vol. 2. IEEE, 2003.
10. Zhao, Tao, Ram Nevatia, and Bo Wu. "Segmentation and tracking of multiple humans in crowded environments." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.7 (2008): 1198-1211.

11. Enzweiler, Markus, and Dariu M. Gavrila. "A mixed generative-discriminative framework for pedestrian classification." *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008.
12. Wu, Bo, and Ramakant Nevatia. "Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors." *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*. Vol. 1. IEEE, 2005.
13. Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE, 2005.
14. Zhu, Qiang, et al. "Fast human detection using a cascade of histograms of oriented gradients." *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. Vol. 2. IEEE, 2006.
15. Maji, Subhansu, Alexander C. Berg, and Jitendra Malik. "Classification using intersection kernel support vector machines is efficient." *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008.
16. Watanabe, Tomoki, Satoshi Ito, and Kentaro Yokoi. "Co-occurrence histograms of oriented gradients for pedestrian detection." *Pacific-Rim Symposium on Image and Video Technology*. Springer Berlin Heidelberg, 2009.
17. Tuzel, Oncel, Fatih Porikli, and Peter Meer. "Pedestrian detection via classification on riemannian manifolds." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.10 (2008): 1713-1727.
18. Schwartz, William Robson, et al. "Human detection using partial least squares analysis." *Computer vision, 2009 IEEE 12th International Conference on*. IEEE, 2009.

19. Oliveira, Luciano, Urbano Nunes, and Paulo Peixoto. "On exploration of classifier ensemble synergism in pedestrian detection." *IEEE Transactions on Intelligent Transportation Systems* 11.1 (2010): 16-27.
20. Walk, Stefan, et al. "New features and insights for pedestrian detection." *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010.
21. Wang, Xiaoyu, Tony X. Han, and Shuicheng Yan. "An HOG-LBP human detector with partial occlusion handling." *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009.
22. Wojek, Christian, Stefan Walk, and Bernt Schiele. "Multi-cue onboard pedestrian detection." *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009.
23. Viola, Paul, Michael J. Jones, and Daniel Snow. "Detecting pedestrians using patterns of motion and appearance." *International Journal of Computer Vision* 63.2 (2005): 153-161.
24. Jones, Michael J., and Daniel Snow. "Pedestrian detection using boosted features over many frames." *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE, 2008.
25. Dalal, Navneet, Bill Triggs, and Cordelia Schmid. "Human detection using oriented histograms of flow and appearance." *European Conference on Computer Vision*. Springer Berlin Heidelberg, 2006.
26. Walk, Stefan, Konrad Schindler, and Bernt Schiele. "Disparity statistics for pedestrian detection: Combining appearance, motion and stereo." *European Conference on Computer Vision*. Springer Berlin Heidelberg, 2010.
27. Munder, Stefan, and Dariu M. Gavrila. "An experimental study on pedestrian classification." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.11 (2006): 1863-1868.

28. Broggi, Alberto, et al. "Model-based validation approaches and matching techniques for automotive vision based pedestrian detection." *Computer Vision and Pattern Recognition-Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on.* IEEE, 2005.
29. Leibe, Bastian, Edgar Seemann, and Bernt Schiele. "Pedestrian detection in crowded scenes." *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on.* Vol. 1. IEEE, 2005.

APPENDIX 1 – Software Manuals

Pedestrian Video Recording Tool Manual

Pedestrian Video Recording Tool Software Manual

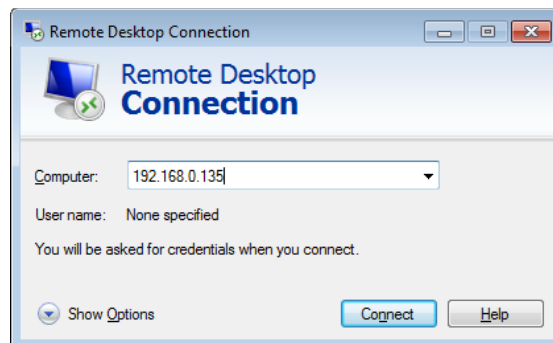
Description

The pedestrian Video Recording Tool (VRT) is the program required for recording video feeds for future processing for the pedestrian detection system. This program requires that the cameras be positioned in their ideal orientations for recording using a 3rd party program as detailed in this document. This tool allows a user to configure the cameras and start recording video in the required format for the Pedestrian Tracker Tool. See the *Pedestrian Tracker Tool Software* manual for details.

Initial Set-up Process

The trailer system contains a recording PC and four high definition AXIS brand IP cameras. Once the trailer has been physically deployed, a connection to the PC is required in order to configure the cameras and initialize the recording. This is accomplished either by connection an monitor and a mouse directly to the PC, or by establishing a remote connection to the PC using a windows laptop, the wifi connection, and the remote desktop application provided by windows.

The trailer IP address is 192.168.0.135, you must use this address to establish the connection in the remote desktop software. Please refer to the windows documentation for usage of the remote desktop program. Once a connection is established, the remote desktop program will ask for a password. The password for the system is "trailer". An image of the Remote Desktop Connection dialog is shown below.



Trailer IP: 192.168.0.135

Password: trailer

Once the trailer/recording system has been properly deployed, and a remote desktop connection to the PC has been established, the cameras must be positioned using the Axis Camera Companion software. In order to start this software, double click the Axis camera Companion shortcut on the desktop. An image of this icon is shown below.

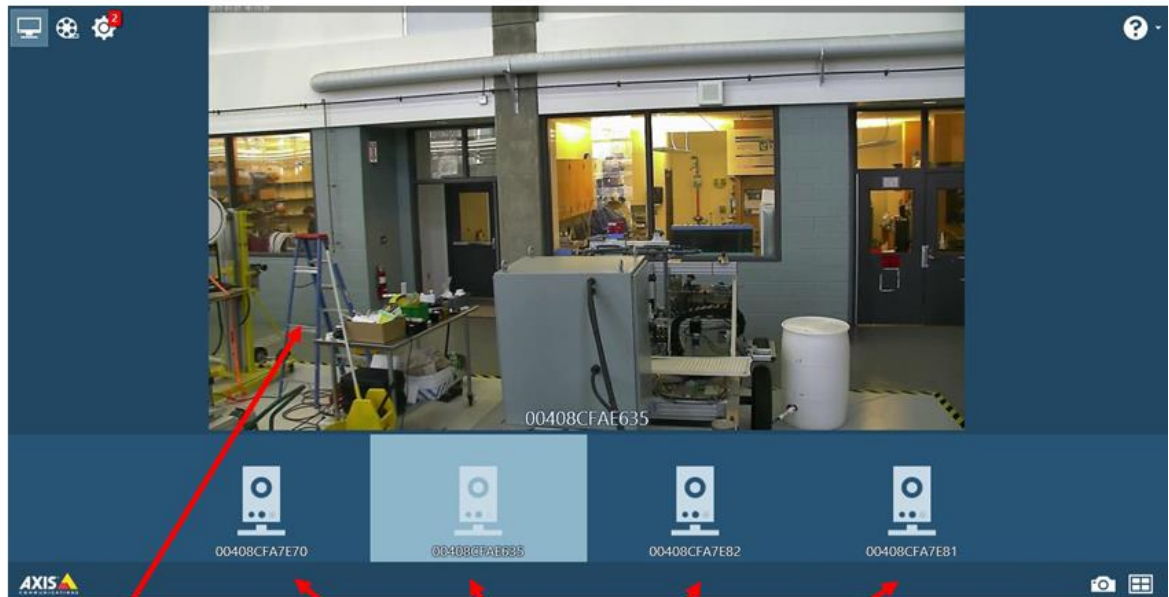


Once the Axis camera Companion application is run, you will be presented with a password request. There is no password, so this can be skipped by pressing the arrow button to the right of the edit box. This will take you to the main dialog for the cameras. The password request dialog with a red arrow indicating the button to press is shown in the image below.



Next you will see a display with all four cameras. This is where the cameras can be positioned and zoomed to the appropriate desired locations. Pick a camera by clicking on one of the four camera icons along the bottom of the dialog. Zooming is achieved using the +/- keys on the keyboard or the scroll wheel on the mouse. Changing where the cameras are pointing is simply performed by clicking in the image where you would like the center of the image to be. This will move the cameras to center them on the point that was clicked in the image. A screenshot of this dialog is shown below. There is extensive documentation on the operation of this program

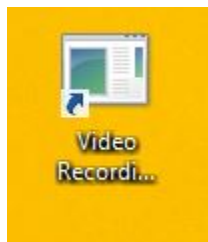
located in the Axis folder from the start menu if necessary.



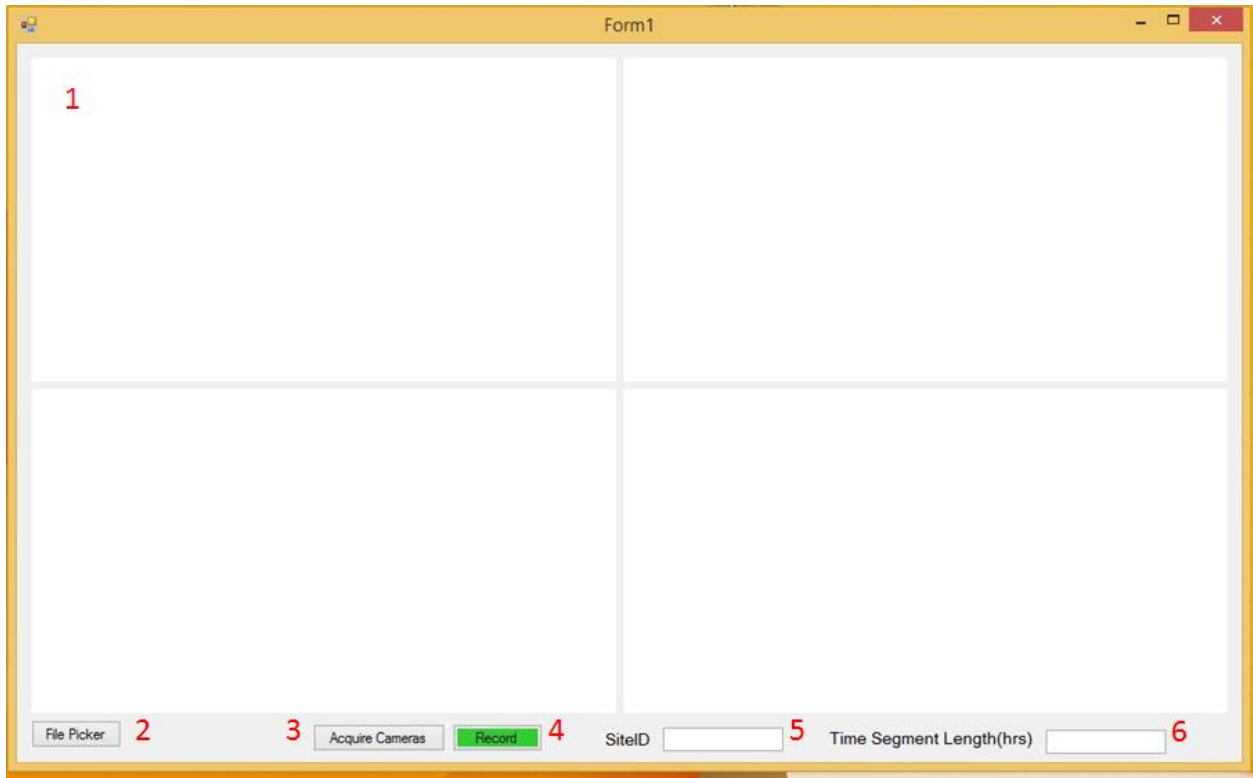
Current Selected Camera View

Camera Selection Buttons

Once all four cameras are positioned appropriately, exit the Axis Camera Companion application and start the VRT. To start the VRT, simply double click the icon on the desktop. The VRT icon is shown below.



VRT – Main Dialog



- 1 – Camera View Windows
- 2 – Save Location Selection Button
- 3 – Acquire Camera Button
- 4 – Record Button
- 5 – Site ID Edit Box
- 6 – Time Segment Length

1 – Camera View Windows

View for displaying the live camera feeds once the cameras are acquired.

2 – Save Location Selection Button

Button for selecting the folder to save video data.

3 – Acquire Camera Button

Button for acquiring and connecting to all IP cameras.

4 – Record Button

Button to start and stop recording.

5 – Site ID Edit Box

Button to set the site identifier associated with the recorded video.

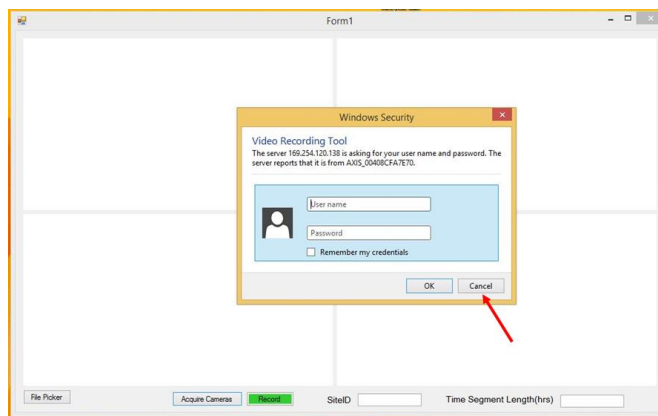
6 – Time Segment Length Edit Box

Edit box for setting the length of time for each individual recorded file in hours. Can accept fractions of an hour. It is suggested not to set this higher than 1 hour.

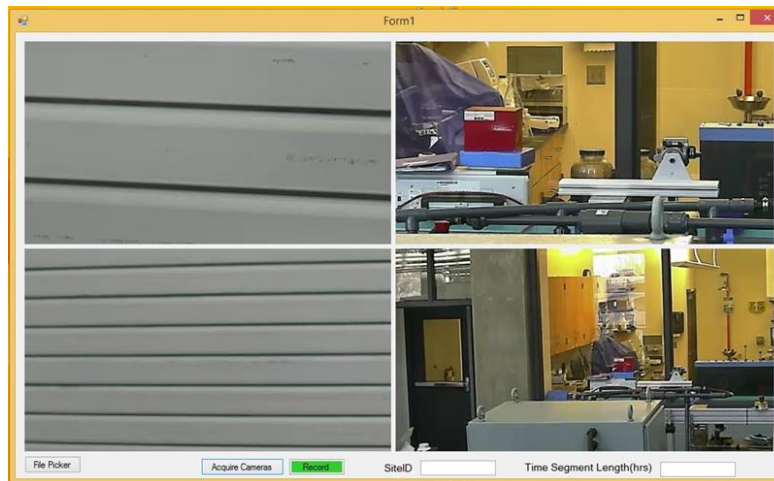
Steps for recording Video

This section describes the steps necessary to connect to the IP cameras and record video data.

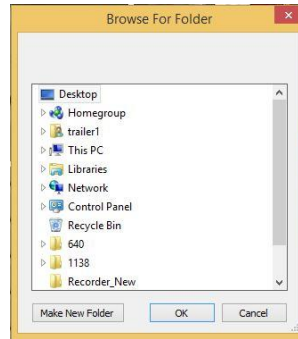
1. Acquire and connect to the IP cameras using the *Acquire Cameras* button (3)
 - a. A dialog will pop up asking for a password. Because there is no password for the IP cameras, press the “Cancel” button as illustrated in the image below to ignore. You may have to do this once for each camera.



2. Press the *Acquire Cameras* button (3) a second time to acquire the cameras. The cameras should appear in the *Camera View Windows* (1) as illustrated below. The camera image will be cropped and only a portion will be shown. The full view should be pre-configured via the Axis Camera Companion application as described in the above section.



3. Select the location to save the images using the *Save Location Selection* button (2). This will open a dialog allowing you to select or create a new directory. This will act as the top-level directory where the files will be saved.



4. Set the Site Identifier using the *Site Identifier* edit box (5). This will create a subdirectory with the name of the site identifier under the directory that was specified in step 3. All videos will be saved to a subfolder for each camera under the main directory. Do not modify or change the directory names as these are required for the Pedestrian Tracking Tool.
5. Set the time segment length of each video clip using the *Time Segment Length* edit box (6). This will set the length of each recorded video clip. The unit is hours and will take fractions of an hour. It is recommended to use between .5 and 1 hour segments, but any segment length can be selected.
6. Press the *Record* button (4) to initiate recording. When recording the button will change from green to red. You can press the button a second time to stop the recording.
- 7.



8. When recording, the *Record* button will turn red and read "Stop". Press the button again to stop the recording. The button will turn green and read "Record" when it has successfully stopped. Recording can be started and stopped as often as desired.

Pedestrian Tracking Tool Manual

Pedestrian Tracker Tool Software Manual

Description

The pedestrian tracker tool is the program for processing raw video data and detecting pedestrians. Their trajectories as they cross the road are recorded to a local database for future analysis. This tool allows a user to specify the video source, configure the image streams, and select which database ID to use for data storage. This manual describes those processes in detail.

User Interface – Main Dialog



- | | |
|----------------------------------|-------------------------|
| 1 – Select Directory Button | 6 – Load Mask Button |
| 2 – Site ID Edit Box | 7 – Camera View Windows |
| 3 – Site Description Edit Box | 8 – Process Button |
| 4 – Camera Selection Check Boxes | 9 – Stop Button |
| 5 – Create Mask Button | 10 – Exit Button |

1 – Select Directory Button

Clicking the *Select Directory Button* will open a dialog asking the user to select the directory containing the raw video data. It is important that the video data is in the format required for processing. This format is automatically created from the Video Recording Tool (see Video Recording Tool Software Document for details). Video files are located in a subfolder named after their site identifier (as assigned in the Video Recording Tool). Each camera has a subfolder containing its respective video files. Please select the top level (Site ID) folder. For example, in the sample file structure in Figure 1, the top level folder that should be selected is named 'Site1'. Site identifier folder names will typically be numeric (example: 001, 002, 003, etc.).

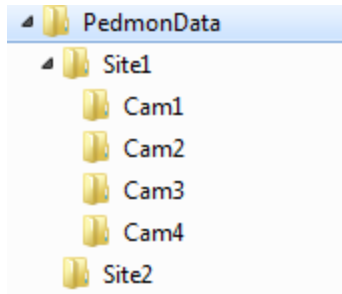


Figure 1- File Structure Sample

Once selected, the dialog will automatically populate the camera views (7) for the selected cameras with the first image from the first video in each camera subfolder. Only cameras that are selected to be processed will have populated views. The dialog will generate an error if the video file structure is incorrect, or if video files are not located properly.

2 – Site ID Edit Box

Enter the Site Identifier number into the *Site ID Edit Box*. This number will be used in the database for all data commitment and future retrieval. This must be the same number as the Site identifier used as the folder name as assigned from the Video Recording Tool.

3 – Site Description Edit Box

Enter an optional text description of the site in this edit box. Typically, this is used for special notes or for containing the address of the site. This description will be logged to the database in the Site ID table to the Site Description field for future retrieval (See the Database Design Document for details on database tables and fields).

4 – Camera Selection Check Boxes

These check boxes will select and de-select which camera videos to process. Only selected (checked) camera video feeds will be processed and results stored in the database. This allows for processing of sites with fewer than four cameras.

5 – Create Mask Button

There is a *Create Mask Button* for each camera. This button will only be enabled if the camera is selected via the associated camera check box. This button will open a dialog for creating a new image mask, which is a required step for processing video data. See details and instructions for the Create Mask Dialog in the following section in this document.

6 – Load Mask Button

Similar to the *Create Mask Button*, this button is only enabled if the associated camera is selected. This button allows for loading of a previously generated mask using the *Create Mask Button*. This is primarily used for reprocessing of data.

7 – Camera View Windows

Once a site is loaded via the *Select Directory Button*, camera views for each of the selected cameras (via the *Camera Selection Check Boxes*) will be shown in these windows. The image shown is loaded from the first frame in the first available video file in each of the associated

camera folders. Once a mask is applied via the *Create Mask* or *Load Mask* buttons, the mask will be drawn onto the image. This allows for visual verification videos before starting processing.

8 – Process Button

The *Process Button* will initiate the pedestrian tracking algorithms on the selected video feeds via the *Camera Selection Check Boxes*. Once processing has initiated, tracker results will be logged to the local database with the given Site ID. Current processing time is 2-to-1, meaning for each hour of video data, it will take approximately 2 hours of processing time. This time is faster if fewer cameras are selected for processing.

9 – Stop Button

This button will interrupt processing.

10 – Exit Button

This button will interrupt any ongoing processing and quit the program. If no processing is currently being performed, it will simply quit the program.

Create Mask Dialog

This dialog is created when the user presses the *Create Mask Button* on the main User Interface. This dialog will load the image shown in the corresponding camera view and allow for creation of an image mask. An image mask is simply a way of defining areas in an image to process, and areas to ignore (do not process). This is required to speed up the algorithms and to remove tracking of trajectories for pedestrians that are not of interest. For example, pedestrians walking on the sidewalk, or crossing in locations that the user wants to ignore.



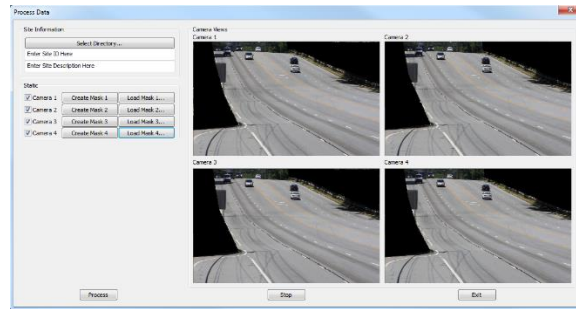
- 1 – Fill Area Button
- 2 – Save Mask Button
- 3 – Undo Button
- 4 – Cancel Button

1 – Fill Area Button

This button will fill the current defined polygon with black pixels (see section Creating a Mask for details). It will be immediately viewable on current image.

2 – Save Mask Button

This button will open a save file dialog to save the mask. Type a filename to save as, and click OK. Upon exit, the mask will be saved to disk and automatically applied to the current camera selection for processing. The User Interface will reflect the changes in the mask in the camera view windows. A sample of the user interface showing masked images ready for processing is shown below. The masked areas are shown as blacked out regions.



3 – Undo Button

This button will remove the last polygon that was filled. Use this if not satisfied with the current mask results. This will only undo the most recent filled polygon.

4 – Cancel Button

This button will close the dialog and discard any changes.

Steps for processing

This section describes the steps necessary to load and process a video.

1. Select the video file directory using the *Select Directory Button* (1)
2. Enter the site ID using the *Site ID Edit Box* (2)
3. Enter the optional site description in the *Site Description Edit Box* (3)
4. Select the Cameras to process with the *Camera Selection Check Boxes* (4)
5. Load or create the image mask using the *Create Mask or Load Mask Buttons* (5, 6). See the *Creating a Mask* section for details.
6. Make sure the images are ready for processing by visualizing the masks and images in the *Camera View Windows* (7)
7. Press the *Process Button* (8) to start processing.

Creating a Mask

This section describes how to create a mask for a camera to prepare for processing. The purpose of creating a mask is foremost to ignore areas of the image where we do not want to track pedestrians such as on sidewalks or other areas that are not of interest for the particular analysis. This section assumes you are already familiar with the dialog and the buttons as described above in this document.

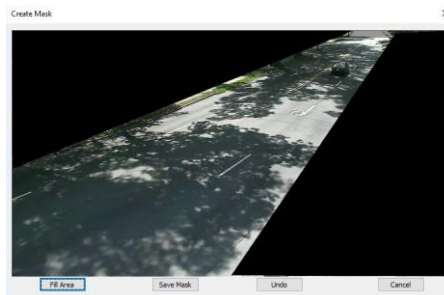
Creating a mask requires filling in polygons in the image with black pixels. The polygons are created by clicking in the image with the left mouse button. Three or more points defines the polygon for filling with pixels. If you make a mistake, you can always use the *Undo Button*.

- 1) Using the mouse, create anchor points by clicking in the image where you want the corners of your shape to fill.

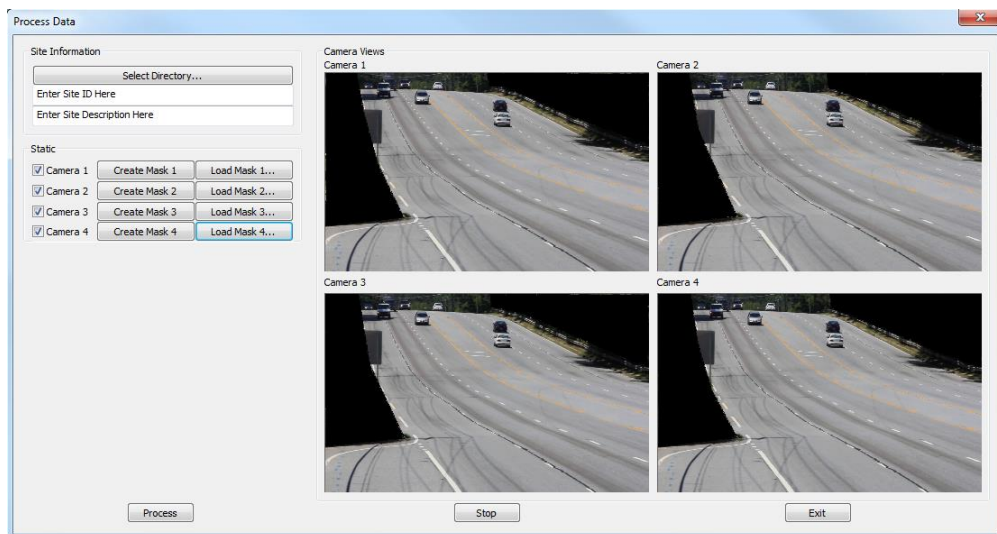
- 2) Press the *Fill Area Button* (1) to fill the area with black pixels. The dialog will update the image with the filled pixels. A sample of a filled polygon is shown below.



- 3) Repeat this process as many times as necessary to fill in all the areas of the image that you do not want to process. An example of a finished mask image is shown below.



- 4) Click the *Save Mask Button* (2) to save the mask and apply it to the camera video stream. Once applied it will be immediately visible in the main user interface as shown below.



- Click the Exit Button if you want to close the dialog and discard the mask. No image mask will be applied to the video stream. You MUST click the *Save Mask Button* (2) to save and apply the mask.

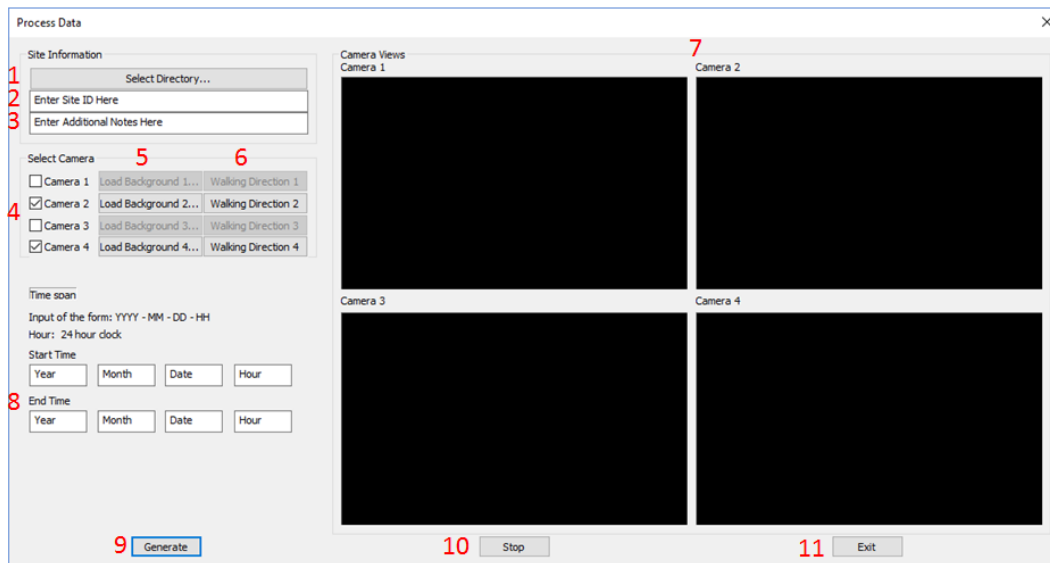
Report Generation Tool Manual

Pedestrian Report Generation Tool Software Manual

Description

The pedestrian Report Generation Tool is the program for generating and viewing results generated from the pedestrian tracking program. This program is designed to be run after completing analysis with the PedMon Tracker Software. See the *PedMon Tracker Software Manual* for details. This tool allows a user to specify the site identification, select which camera's to query, and select the time period for reporting. This manual describes those processes in detail.

User Interface – Main Dialog



- | | |
|----------------------------------|--------------------------------------|
| 1 – Select Directory Button | 6 – Walking Direction Button |
| 2 – Site ID Edit Box | 7 – Camera View Windows |
| 3 – Additional Notes Edit Box | 8 – Enter Start/Stop Time Edit Boxes |
| 4 – Camera Selection Check Boxes | 9 – Generate Report Button |
| 5 – Load Background Button | 10 – Stop Button |
| | 11 – Exit Button |

1 – Select Directory Button

Clicking the *Select Directory Button* will open a dialog asking the user to select the directory containing the raw video data. It is important that the video data is in the format required for processing. This format is automatically created from the Video Recording Tool (see Video Recording Tool Software Document for details). Video files are located in a subfolder named after their site identifier (as assigned in the Video Recording Tool). Each camera has a subfolder containing their respective video files. Please select the top level (Site ID) folder. For example, in the sample file structure in Figure 1, the top level folder that should be selected is named 'Site1'. Site identifier folder names will typically be numeric (example: 001, 002, 003, etc.).

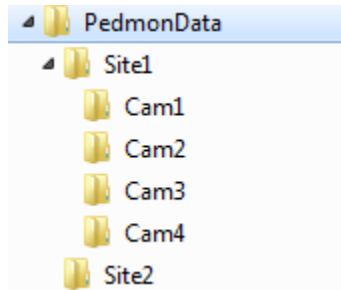


Figure 1- File Structure Sample

Once selected, the dialog will automatically populate the camera views (7) for the selected cameras with the first image from the first video in each camera subfolder. Only cameras that are selected to be processed will have populated views. The dialog will throw an error if the video file structure is incorrect, or if video files are not located properly.

2 – Site ID Edit Box

Enter the Site Identifier number into the *Site ID Edit Box*. This number will be used in the database for all data retrieval. This must be the same number as the Site identifier used as the folder name as assigned from the Video Processing Tool.

3 – Additional Notes Edit Box

Use this edit box to enter any notes that may be pertinent to this report. Notes entered here will be printed into the resulting report file generated.

4 – Camera Selection Check Boxes

These check boxes will select and de-select which camera trajectories to report. Only selected (checked) camera video feeds will be processed and results generated. This allows report generating for sites with fewer than four cameras.

5 – Load Background Button

Report generation requires an image from the site in order to properly visualize pedestrian crossings. If video files are not available as described above using the *Select Directory* button, an alternate method is to load the image manually. This requires that an image from the video data exists. Pressing this button will open a dialog allowing the user to select the image for drawing. The image will immediately show up in the associated camera view window for the selected camera.

6 – Select Walking Direction Button

There is a *Select Walking Direction Button* for each camera. This button will only be enabled if the camera is selected via the associated camera check box. This button will open a dialog for selecting the slope of the direction of pedestrian crossing, which is a required step for generating report data. See details and instructions for the Select Walking Direction Dialog in the following section in this document.

7 – Camera View Windows

Once a site is loaded via the *Select Directory Button*, camera views for each of the selected cameras (via the *Camera Selection Check Boxes*) will be shown in these windows. The image shown is loaded from the first frame in the first available video file in each of the associated camera folders. Once the pedestrian crossing direction is applied via the *Select Walking Direction* button, the slope of the crossing will be drawn onto the image. This allows for visual verification videos before starting processing.

8 – Enter Start/Stop Time Edit Boxes

These edit boxes allow the user to enter the start time and stop time for report generation. Pedestrian trajectories will only be reported and visualized for the time period between the start and end time as specified in the start and stop time edit boxes.

9 – Generate Button

The *Generate Button* will initiate the report generating algorithms on the selected video feeds via the *Camera Selection Check Boxes*. Once processing has initiated, tracker results will be logged to the local database with the given Site ID. Current processing time is 2-to-1, meaning for each hour of video data, it will take approximately 2 hours of processing time. This time is faster if fewer cameras are selected for processing.

10 – Stop Button

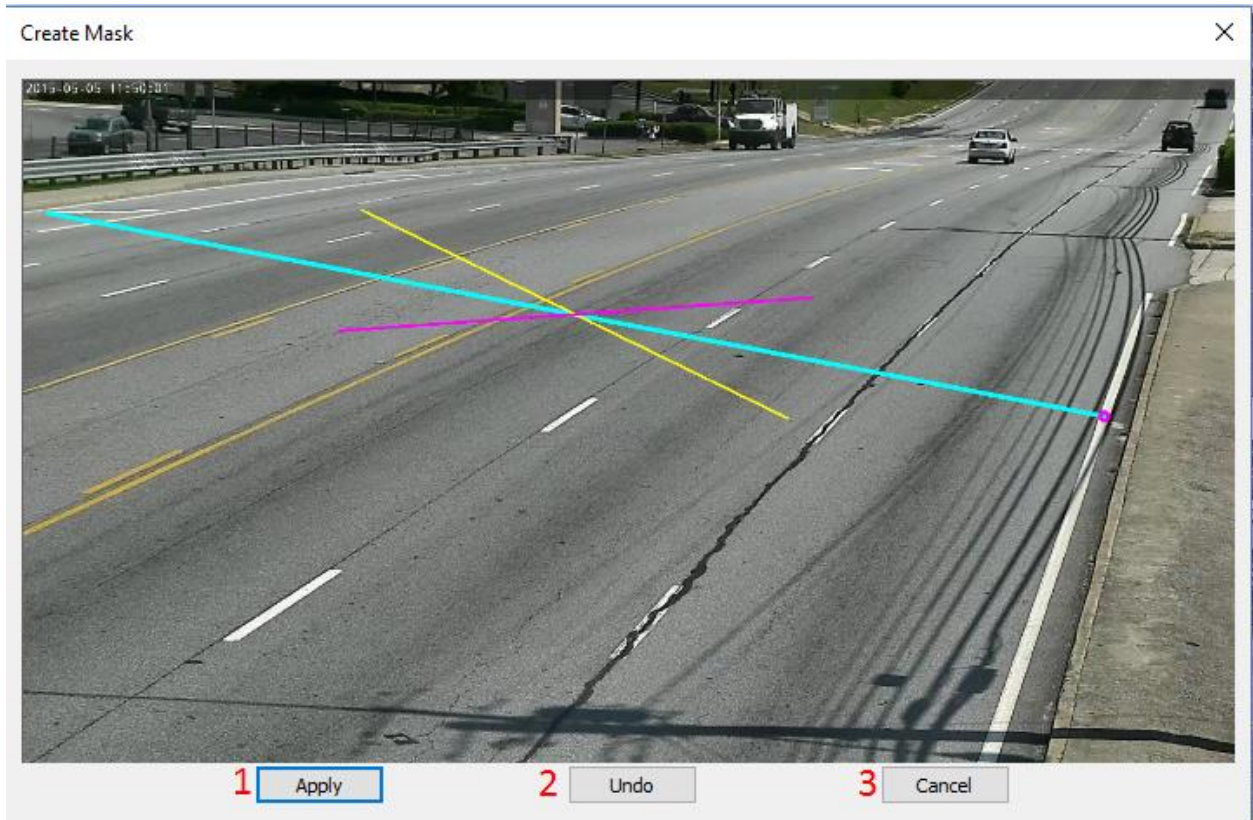
This button will interrupt report generation.

11 – Exit Button

This button will interrupt any ongoing report and quit the program. If no generation is currently being performed, it will simply quit the program.

Select Walking Direction Dialog

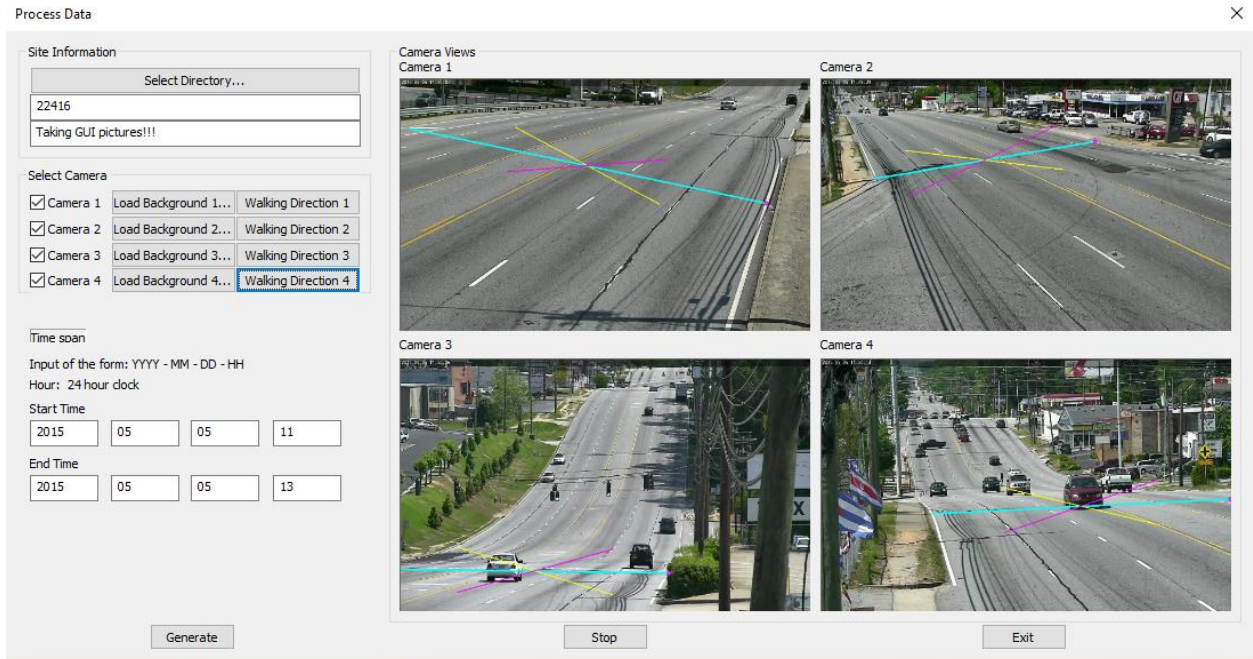
This dialog is created when the user presses the *Walking Direction Button (6)* on the main User Interface. This dialog will load the image shown in the corresponding camera view and allow for selection of the direction of travel of pedestrians crossing the road. This is required to filter out invalid trajectories as a result of the tracking algorithms falsely identifying vehicles and tracking them. This is achieved by defining the slope of a line by selecting two points on the road in the direction of pedestrians crossing. Trajectories that are perpendicular to this slope will be rejected for reporting purposes. This effectively filters out the false trajectories due to vehicles.



- 1 – Apply Button
- 2 – Undo Button
- 3 – Cancel Button

1 – Apply Button

This button will finalize the slope from the points selected by the user and return to the main dialog. The User Interface will reflect the changes in slope mask in the camera view windows. A sample of the user interface showing images with slopes ready for reporting is shown below. The slopes are shown as blue lines in the image.



2 – Undo Button

This button will remove the last slope that was defined. Use this if you are not happy with the current slope results.

3 – Cancel Button

This button will close the dialog and discard any changes.

Steps for generating a report

This section describes the steps necessary to load and process a video.

1. Select the video file directory using the *Select Directory Button* (1), or alternately, load the individual images using the *Load Background* buttons for each camera (6)
2. Enter the site ID using the *Site ID Edit Box* (2)
3. Select the Cameras to process with the *Camera Selection Check Boxes* (4)
4. Create the pedestrian crossing direction slope using the *Road Direction Button* (5). See the *Select Walking Direction Section* below for details.
5. Make sure the images are ready for processing by visualizing the slopes and images in the *Camera View Windows* (7)
6. Enter the start and stop times in the edit boxes (8)
7. Press the *Generate Button* (9) to start report generation.

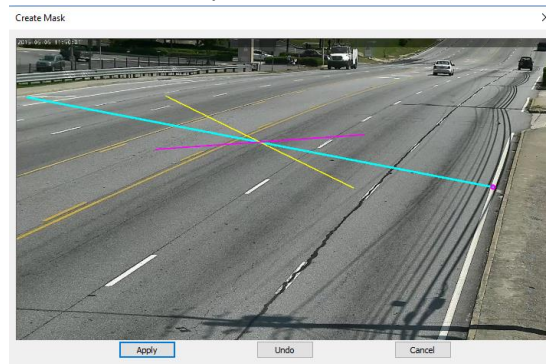
Selecting the Walking Direction

This section describes how to select the pedestrian crossing/walking direction for a camera to prepare for report generation. The purpose of creating a slope for the crossing direction is foremost to filter out incorrect trajectories from vehicles travelling in the direction of traffic.

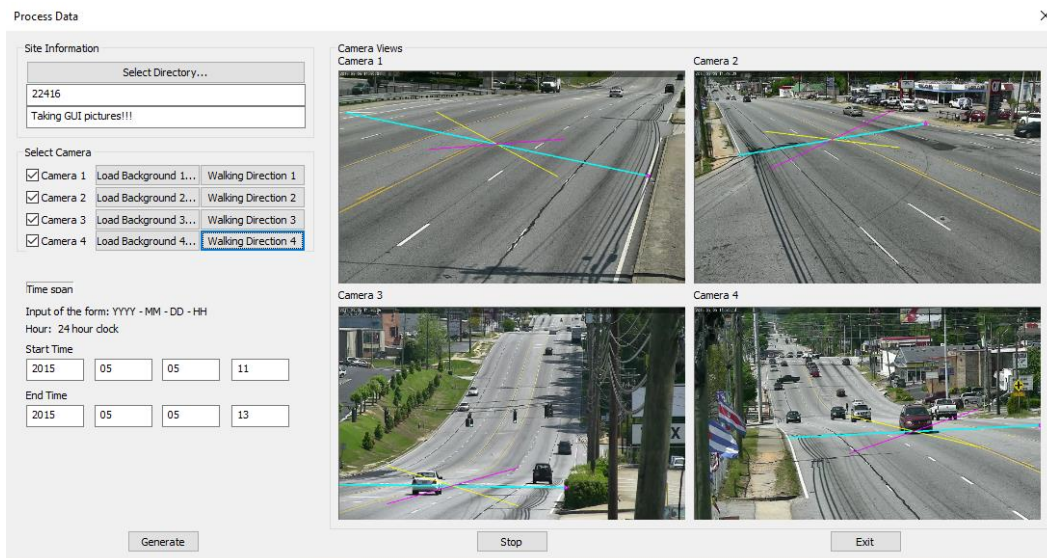
This section assumes you are already familiar with the dialog and the buttons as described above in this document.

Selecting the walking direction requires selecting two points in each image to define the start and stop of a line in the direction of the pedestrian crossing. If you make a mistake, you can always use the *Undo* Button (2).

- 1) Using the mouse, create anchor points by clicking in the image where you want the start and stop points of the line to be.
- 2) Once two points are selected, a line will automatically be drawn into the image indicating the selected crossing direction as shown below. If the slope is incorrect, press the *Undo* button (2) to remove the slope and start over.



- 3) Click the *Apply* Button (1) to save the slope and apply it to the camera video stream. Once applied it will be immediately visible in the main user interface as shown below.



- 4) Click the *Exit* Button if you want to exit the dialog and discard the selected slope.

APPENDIX 2 – Clairemont

Road Analysis Report

Clairemont and Maedris Comprehensive Pedestrian/Cyclist Counts

Details

Video data was collected for five consecutive days from July 6, Wednesday, through July 10, Sunday. Analysis was performed on the pedestrian and cyclist counts between the hours of 6 a.m. and 10 p.m.

Cyclists were not counted if they were in the driving lanes; only cyclists on the sidewalk and through the intersections were recorded. There were considerably higher cyclist counts if you consider the bikes travelling in the vehicle lanes.

Data is presented in both graphical and time-line format. The graphical presentation is for summarizing the counts, while the time-line allows for visualization of the pedestrian frequency over time.

A description of the site is presented next and followed by sections summarizing the data day-by-day. Finally, a summary of the entire 5-day count is detailed.

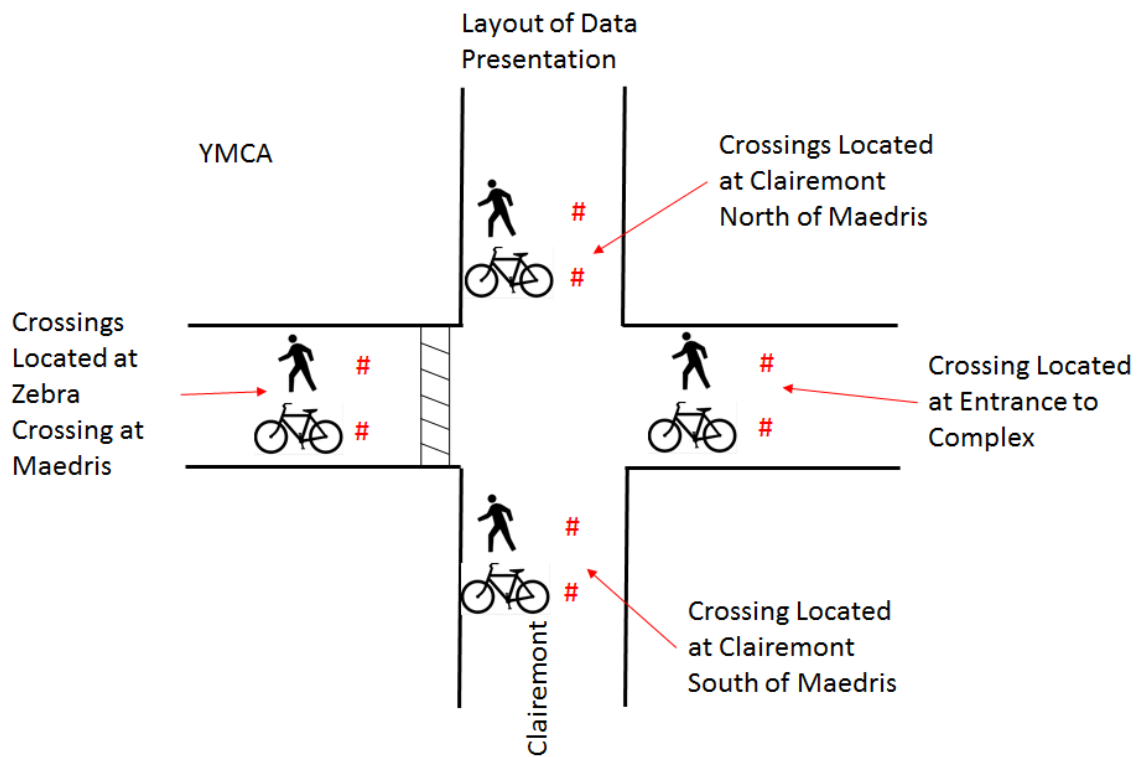
Site Description

The site consisted of a 3-way intersection with Maedris and Clairemont Road. Directly across from Maedris was an entrance to an apartment complex. An existing zebra crossing is present at the intersection on the Maedris side. There are no painted crossings across Clairemont, nor across the entrance to the apartment complex. Counts are presented for all crossings, although the counts for the already painted intersection and the entrance to the complex are not of particular concern. The primary concern is related to crossings across Clairemont where there is no existing markings. See the following figure for an overhead view of the intersection (Source: Google earth).



Figure 1 - Maedris and Clairemont Intersection

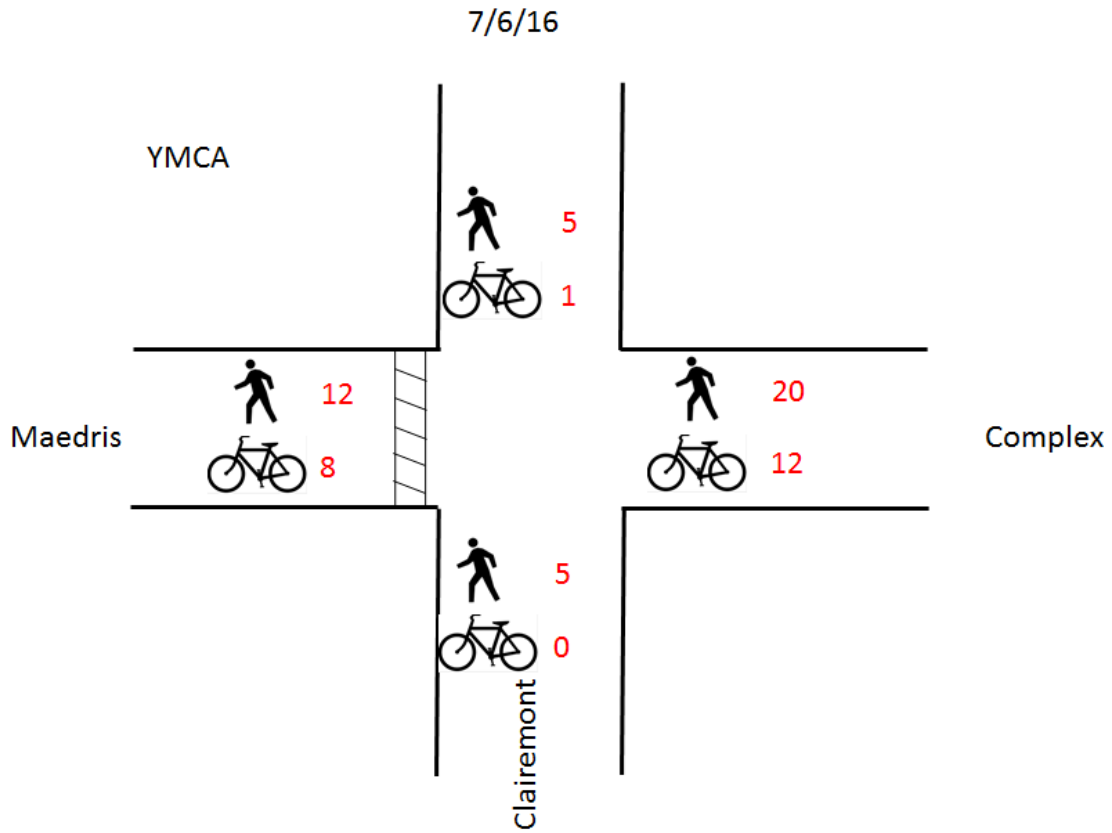
The following figure illustrates the layout of the graphical presentation and the time-lines. The critical crossings are the crossings located on Clairemont to the North and South of Maedris.



Wednesday – 7/6/16

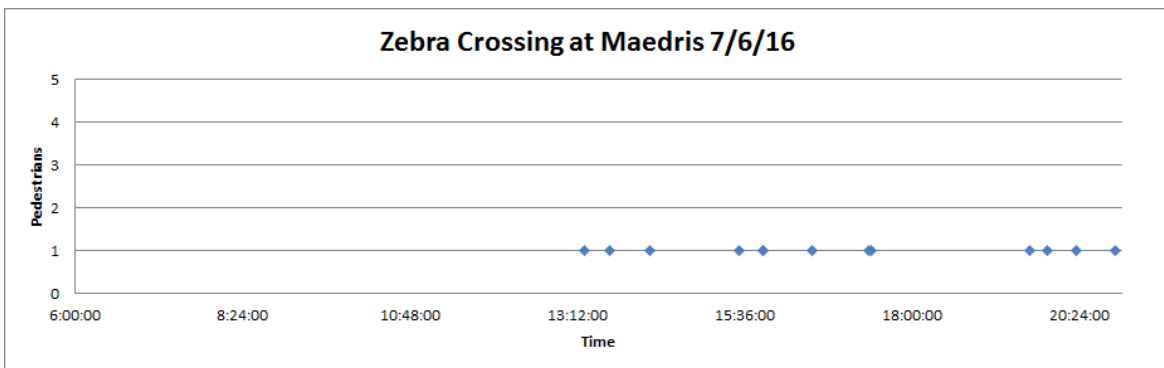
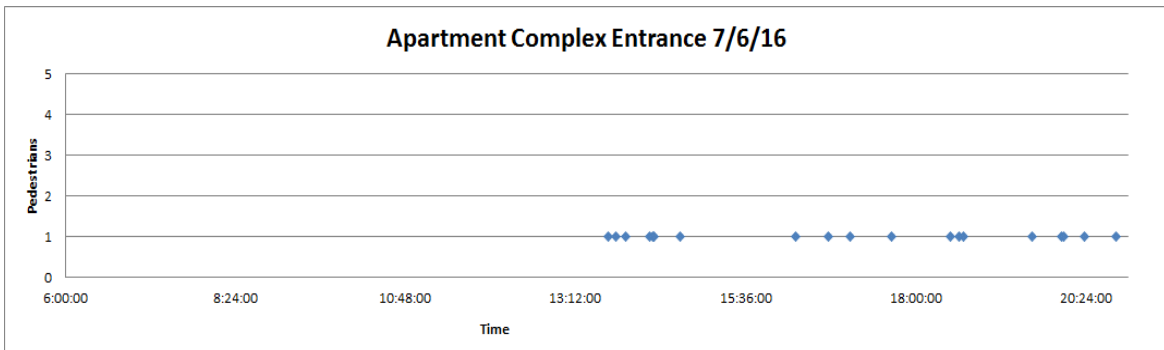
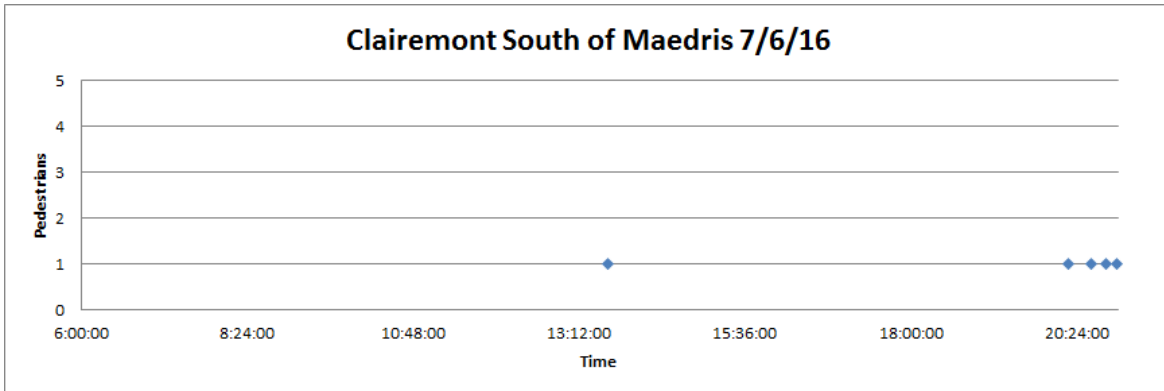
On Thursday, data collection did not begin until 1 p.m. due to system setup. Therefore, the counts are lower on this day than the other days.

Summary Graphic

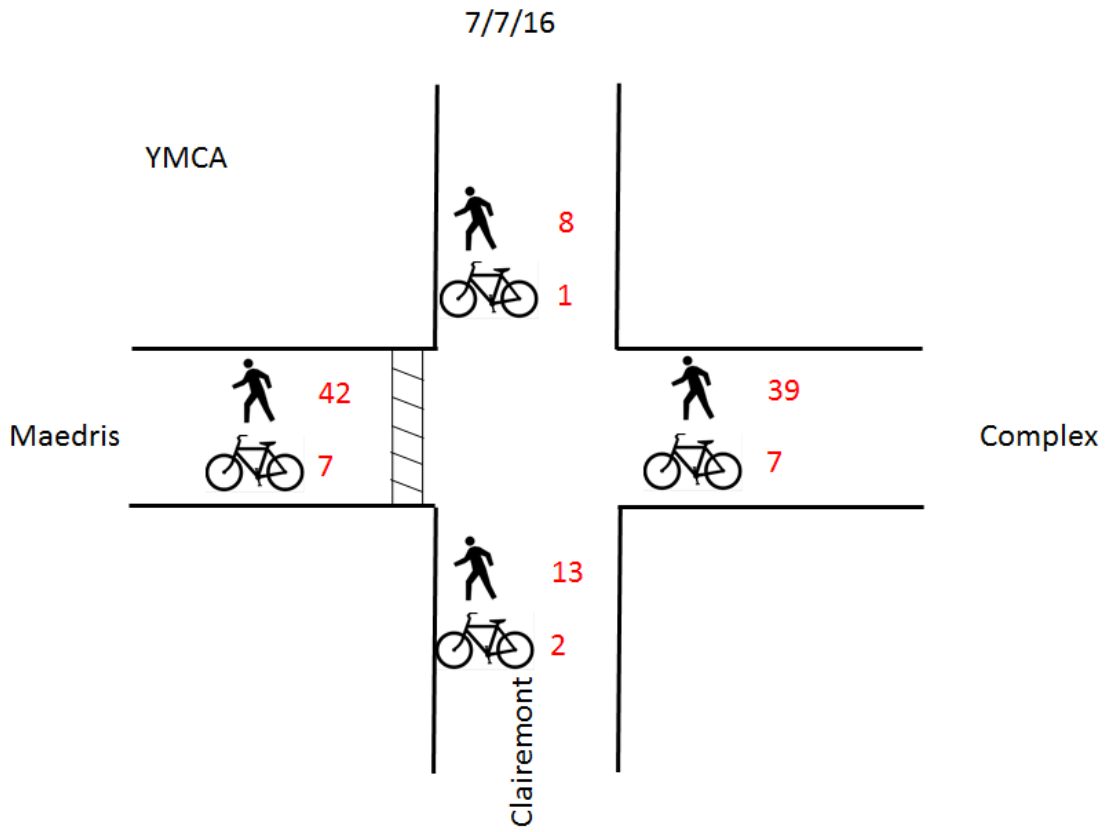


Pedestrian Crossing Time-Line

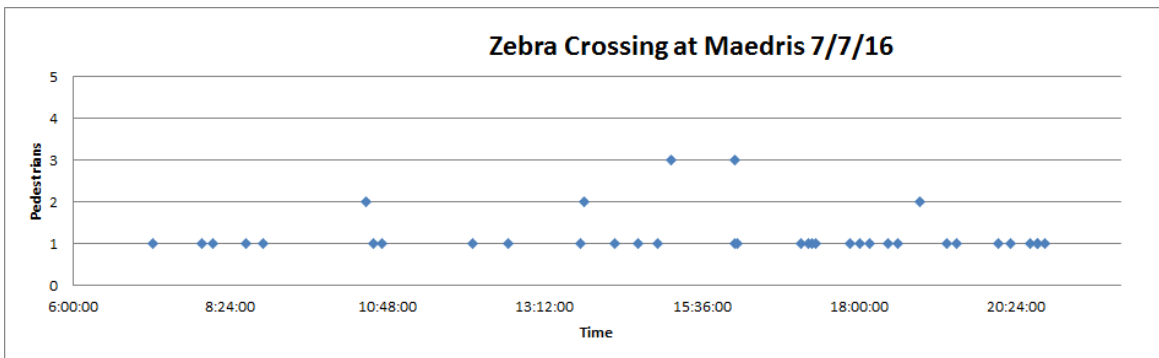
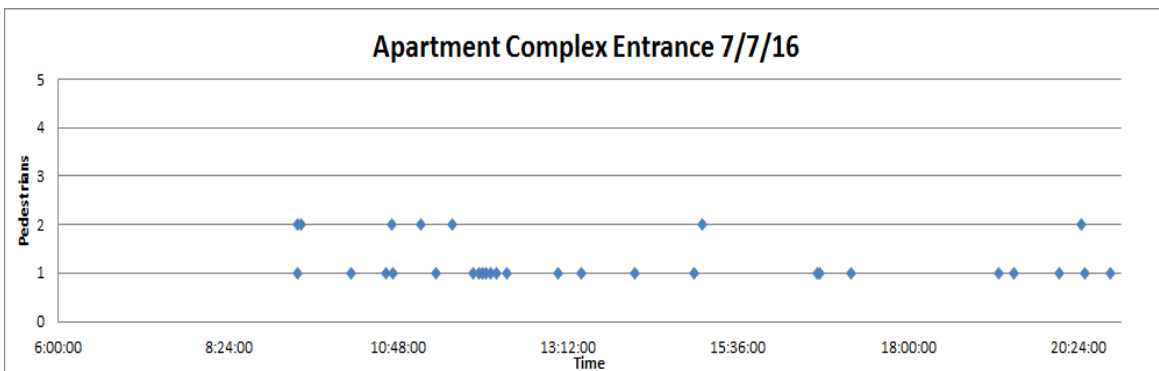
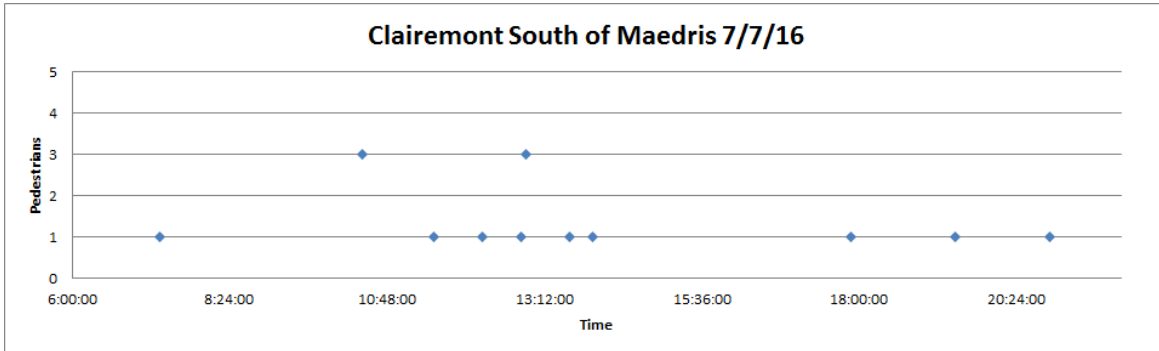
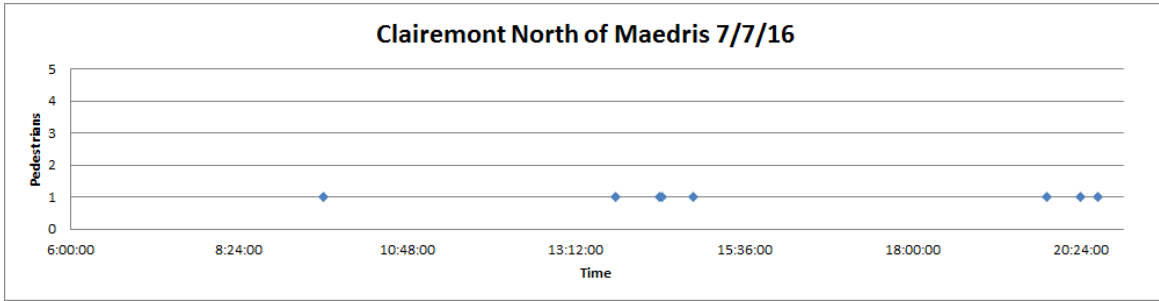
This section details the frequency of crossings on a time-line. This allows for visualization of the timing and frequency of crossings for each location. The pedestrian time-line for Clairemont North of Maedris is incomplete, so it is not presented for this day. It is complete and presented on all following days.



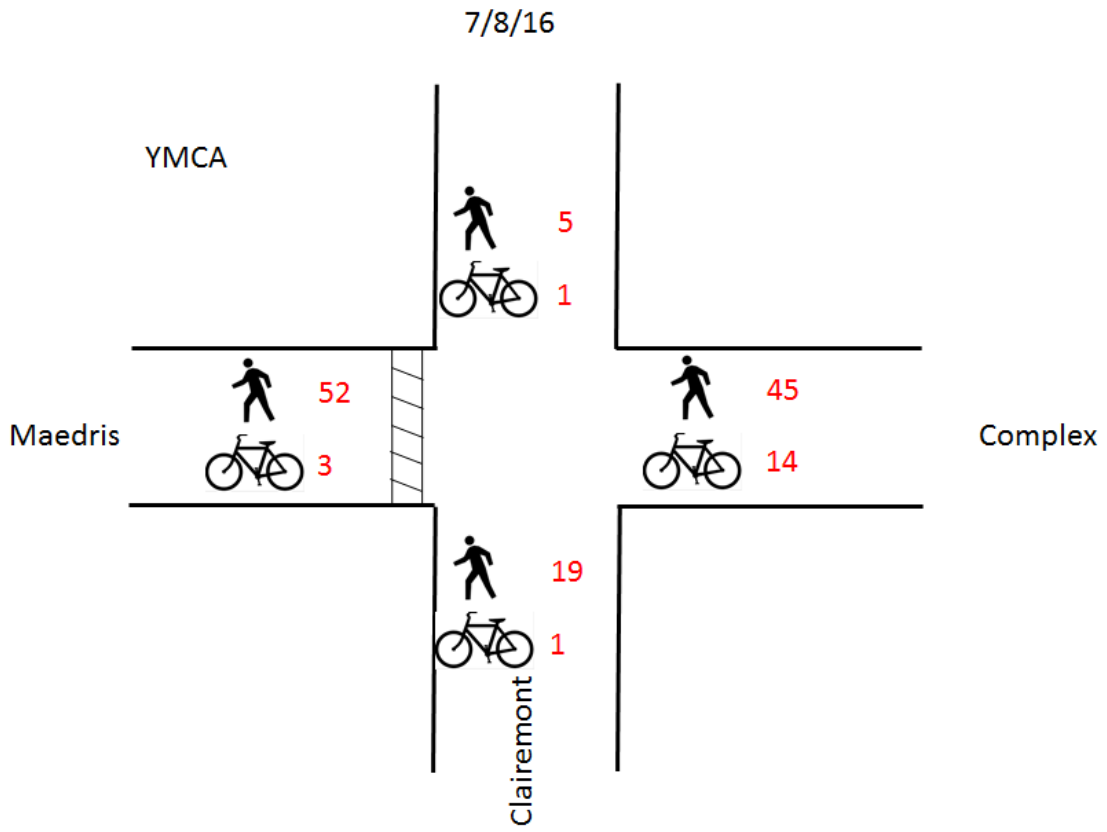
Thursday – 7/7/16
Summary Graphic



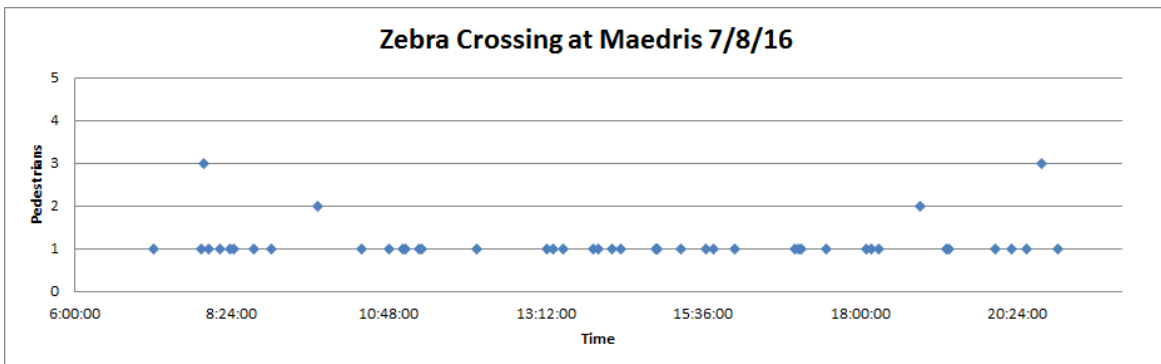
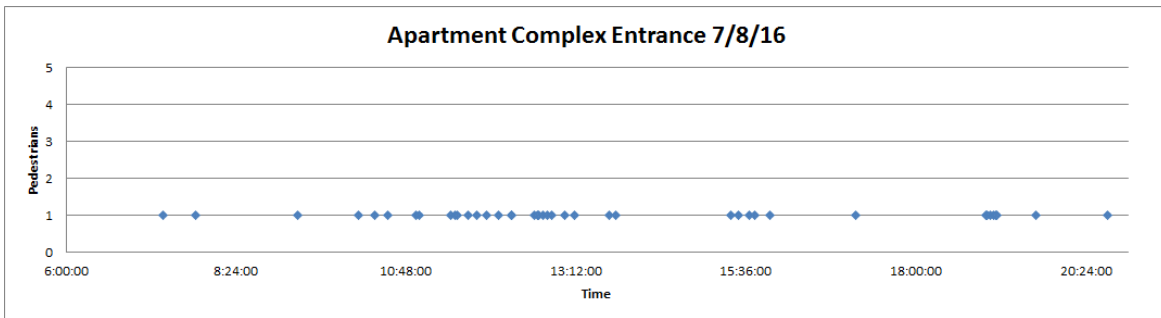
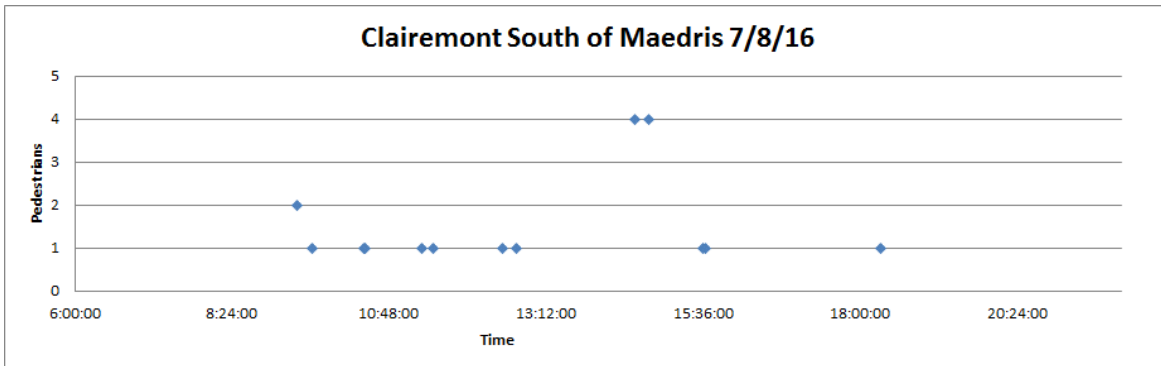
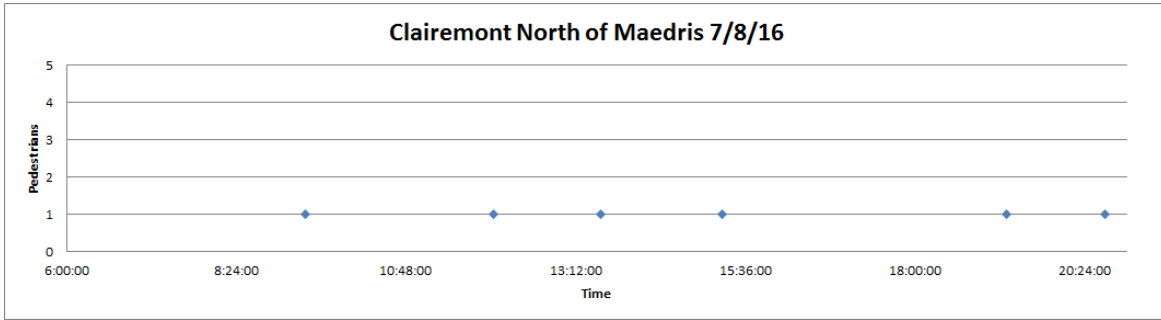
Pedestrian Crossing Time-Line



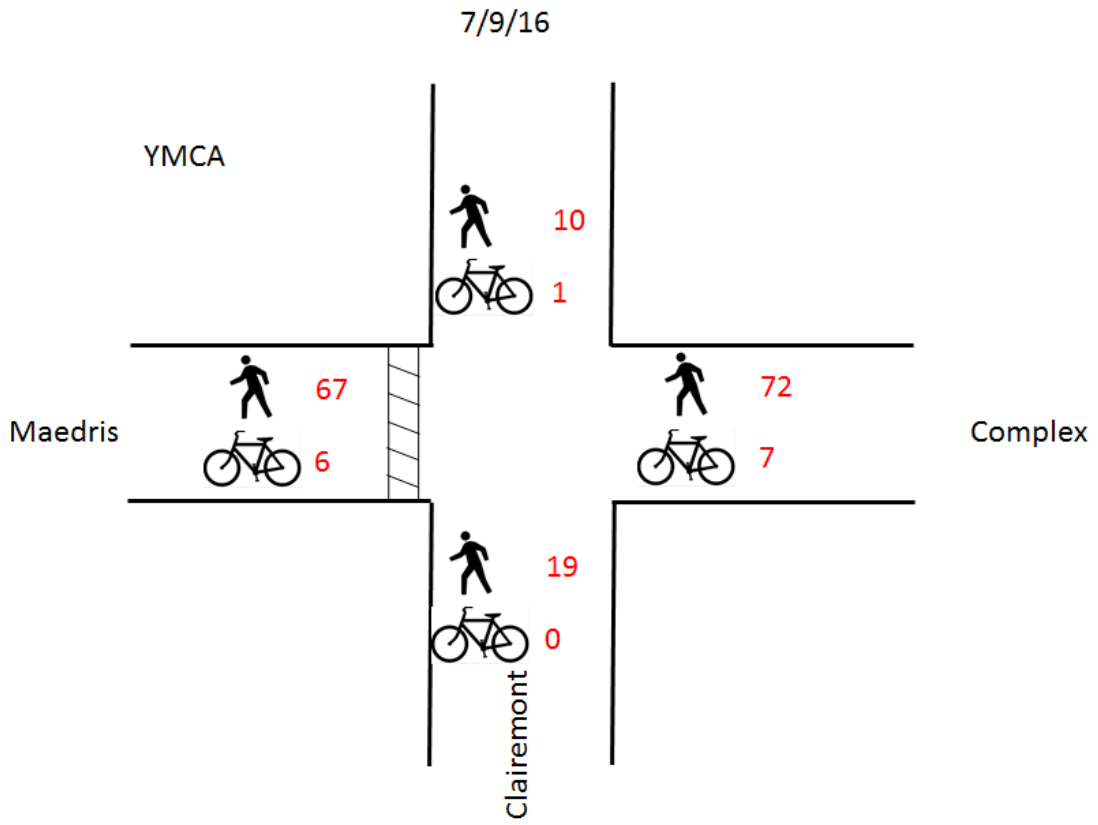
Friday – 7/8/16
Summary Graphic



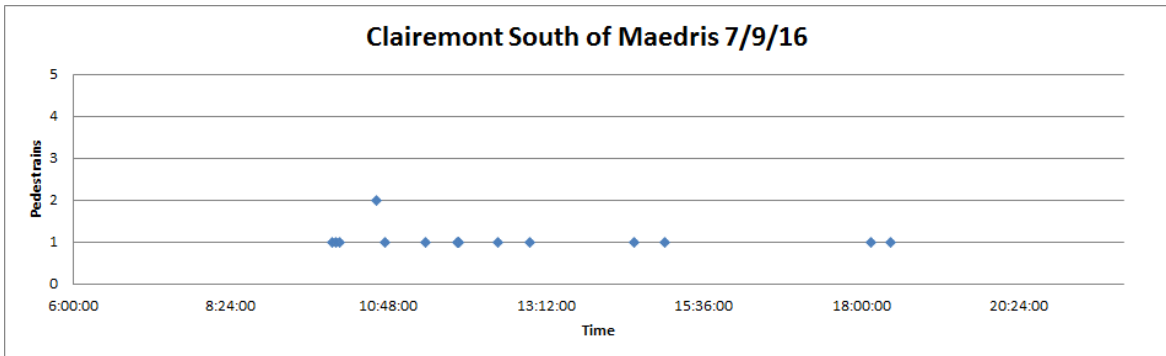
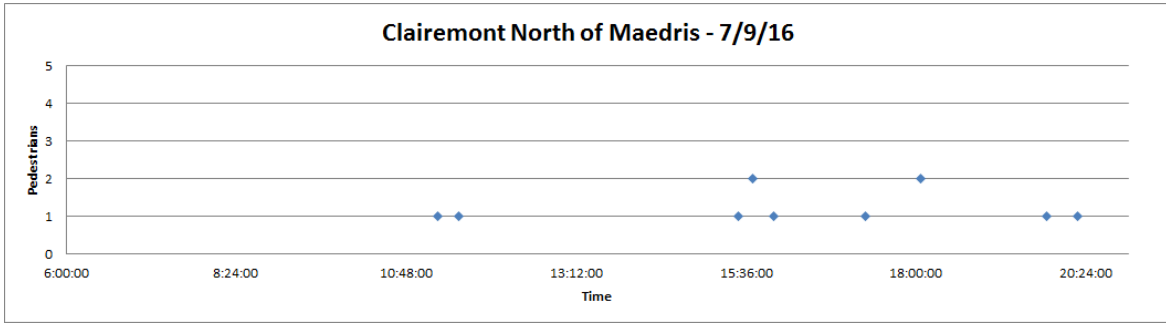
Pedestrian Crossing Time-Line



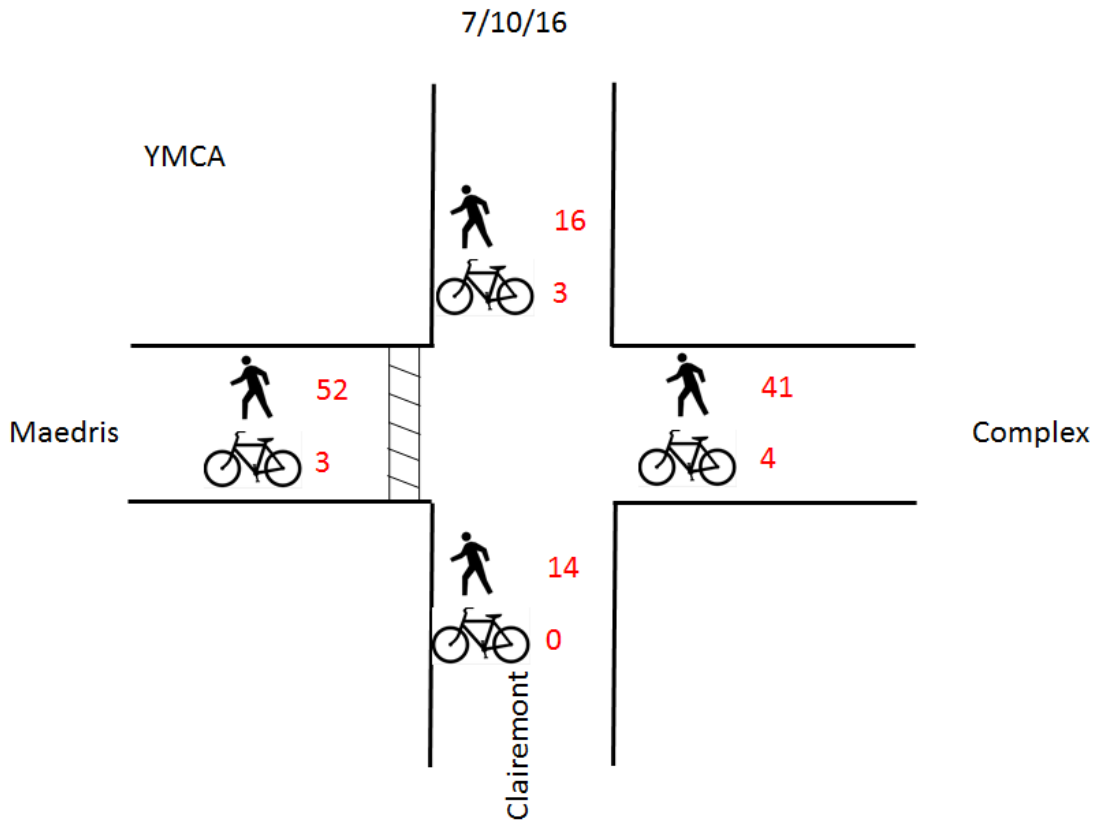
Saturday – 7/9/16
Summary Graphic



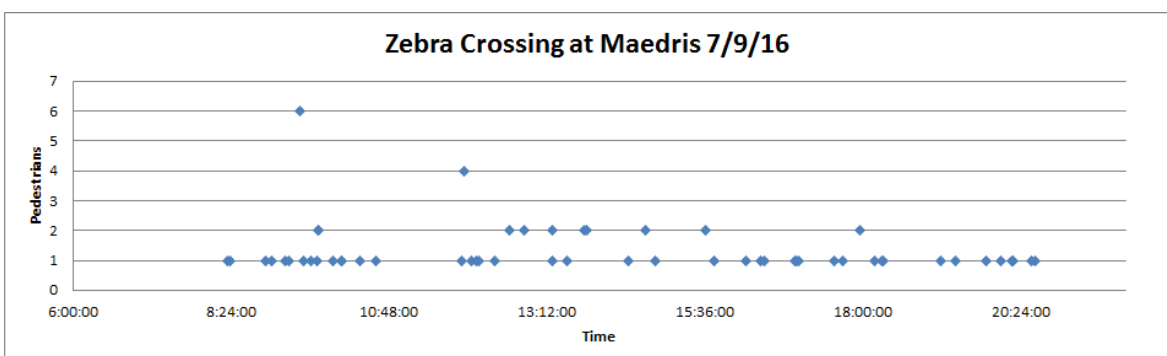
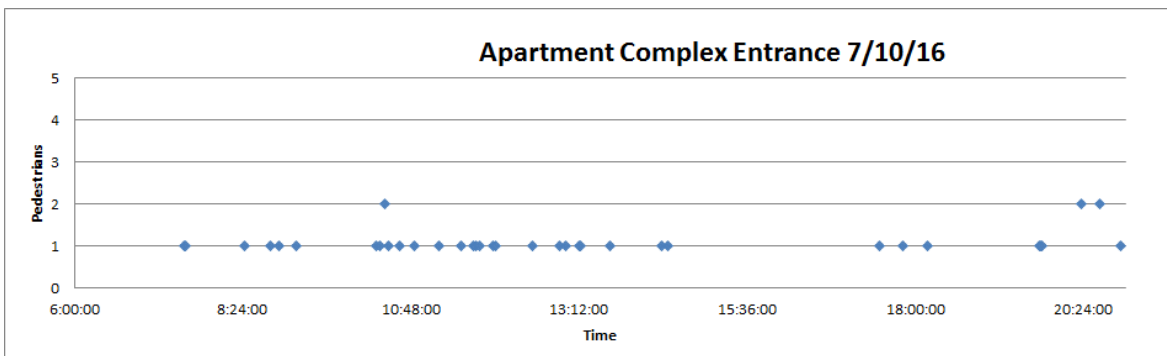
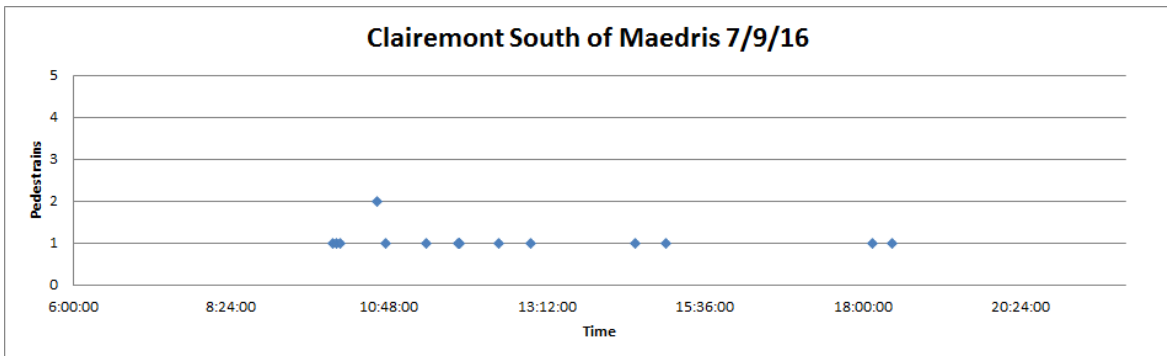
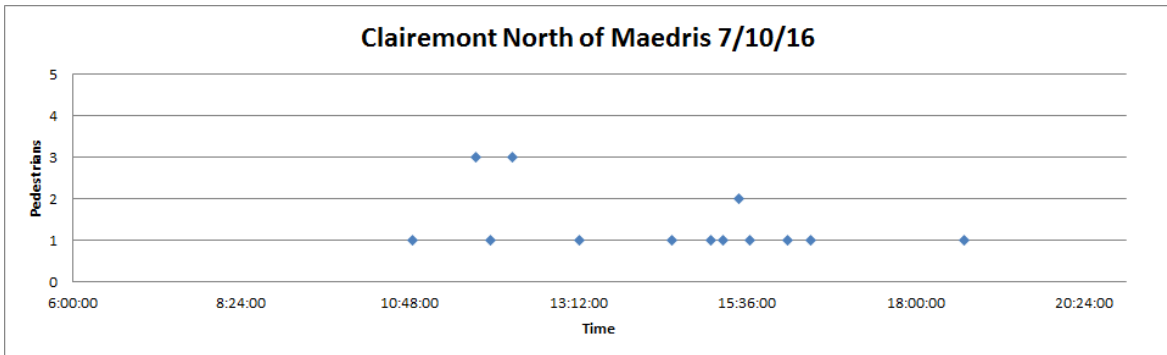
Pedestrian Crossing Time-Line



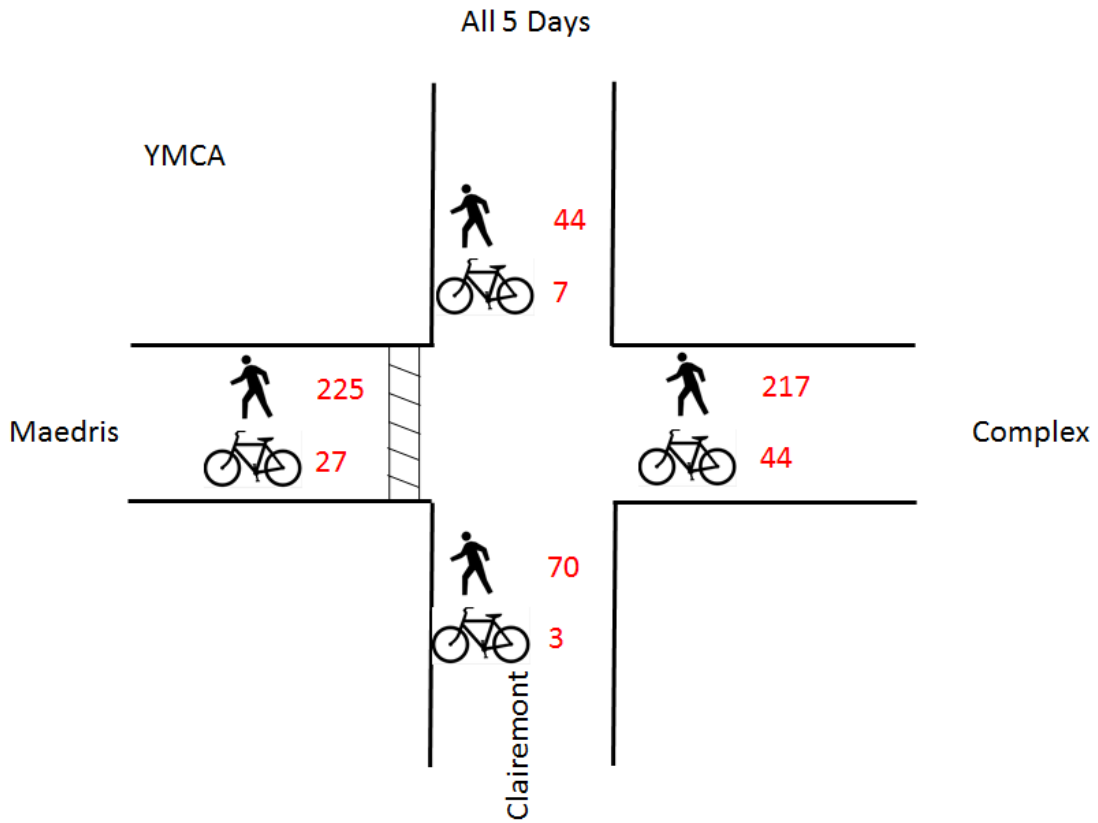
Sunday – 7/10/16
Summary Graphic



Pedestrian Crossing Time-Line



Summary Graphic for All Days



Summary Table for All Days (Pedestrian Only)

DAY	Clairemont North	Clairemont South	Complex entrance	Zebra@Maedris
7/6/16	5	5	20	12
7/7/16	8	13	39	42
7/8/16	5	19	45	52
7/9/16	10	19	72	67
7/10/16	16	14	41	52
All Days	44	70	217	225