Layered Object Recognition System for Pedestrian Sensing

PUBLICATION NO. FHWA-HRT-11-056

OCTOBER 2012

U.S. Department of Transportation
Federal Highway Administration

Research, Development, and Technology
Turner-Fairbank Highway Research Center
6300 Georgetown Pike
McLean, VA 22101-2296

FOREWORD

The purpose of this report is to describe the work performed and the results obtained during the Layered Object Recognition System for Pedestrian Sensing Project sponsored by the Federal Highway Administration. The goal of this project was to use stereo vision to detect, classify, and track pedestrians in cameras' field of views and demonstrate the system's performance in real time in a test vehicle.

This report will be of interest to researchers, developers, and technologists in the area of highway safety, pedestrian collision warning systems, intelligent transportation systems, and driver assistance systems. It provides information about state-of-the-art practices and directions for future work.

Monique R. Evans
Director, Office of Safety
Research and Development

TECHNICAL REPORT DOCUMENTATION PAGE

1. Report No. FHWA-HRT-11-056	2. Government Accession No.	3. Recipient Catalog No.
4. Title and Subtitle Layered Object Recognition System for Pedestrian Sensing		5. Report Date October 2012
		6. Performing Organization Code
7. Author(s) Jayan Eledath, Bogdan Matei, Mayank Bansal, Sang-Hack Jung, and Harpreet Sawhney		8. Performing Organization Report No.
9. Performing Organization Name and Address Sarnoff Corporation 201 Washington Road Princeton, NJ 08543		10. Work Unit No. (TRAIS)
		11. Contract or Grant No. Contract No. DTFH61-07-H-00039
12. Sponsoring Agency Name and Address Exploratory Advanced Research Program Federal Highway Administration 6300 Georgetown Pike McLean, VA 22101-2296		13. Type of Report and Period Covered Final Report
		14. Sponsoring Agency Code

15. Supplementary Notes
Wei Zhang was the Agreement Officer's Technical Representative (AOTR) for the Federal Highway Administration. Technical panel members included Ann Do, David Gibson, John Harding, and Jennifer Percer.

16. Abstract
There is a significant need to develop innovative technologies to detect pedestrians or other vulnerable road users at designated crossing locations and midblock/unexpected areas and to determine potential collisions with pedestrians. An in-vehicle pedestrian sensing system was developed to address this specific problem. The research team used stereo vision cameras and developed three key innovations, namely, the detection and recognition of multiple roadway objects; the use of multiple cues (depth, motion, shape, and appearance) to detect, track, and classify pedestrians; and the use of contextual information to reject a majority of the typical false positives that plague vision-based pedestrian detection systems. This report describes the approach and tabulates representative results of experiments conducted on multiple video sequences captured over the course of the project. The conclusion derived from these results is that the developed system is state of the art when compared to the best approaches published in literature. The false positive rates are still higher than desired for the system to be ready for commercialization. This report also provides steps that can be taken to improve the performance in this regard. A real-time system was developed and demonstrated in a test vehicle.

17. Key Words Pedestrian Safety, Pedestrian detection, Stereo vision, Disparity map, Histogram of oriented gradients (HOG), Contour-based classifier	18. Distribution Statement No restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161

19. Security Classification (of this report) Unclassified	20. Security Classification (of this page) Unclassified	21. No. of Pages 113	22. Price

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

SI* (MODERN METRIC) CONVERSION FACTORS

APPROXIMATE CONVERSIONS TO SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
in	inches	25.4	millimeters	mm
ft	feet	0.305	meters	m
yd	yards	0.914	meters	m
mi	miles	1.61	kilometers	km
		AREA		
in^2	square inches	645.2	square millimeters	mm^2
ft^2	square feet	0.093	square meters	m^2
yd^2	square yard	0.836	square meters	m^2
ac	acres	0.405	hectares	ha
mi^2	square miles	2.59	square kilometers	km^2
		VOLUME		
fl oz	fluid ounces	29.57	milliliters	mL
gal	gallons	3.785	liters	L
ft^3	cubic feet	0.028	cubic meters	m^3
yd^3	cubic yards	0.765	cubic meters	m^3
		NOTE: volumes greater than 1000 L shall be shown in m^3		
		MASS		
oz	ounces	28.35	grams	g
lb	pounds	0.454	kilograms	kg
T	short tons (2000 lb)	0.907	megagrams (or "metric ton")	Mg (or "t")
		TEMPERATURE (exact degrees)		
^oF	Fahrenheit	5 (F-32)/9 or (F-32)/1.8	Celsius	^oC
		ILLUMINATION		
fc	foot-candles	10.76	lux	lx
fl	foot-Lamberts	3.426	$candela/m^2$	cd/m^2
		FORCE and PRESSURE or STRESS		
lbf	poundforce	4.45	newtons	N
lbf/in^2	poundforce per square inch	6.89	kilopascals	kPa

APPROXIMATE CONVERSIONS FROM SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
mm	millimeters	0.039	inches	in
m	meters	3.28	feet	ft
m	meters	1.09	yards	yd
km	kilometers	0.621	miles	mi
		AREA		
mm^2	square millimeters	0.0016	square inches	in^2
m^2	square meters	10.764	square feet	ft^2
m^2	square meters	1.195	square yards	yd^2
ha	hectares	2.47	acres	ac
km^2	square kilometers	0.386	square miles	mi^2
		VOLUME		
mL	milliliters	0.034	fluid ounces	fl oz
L	liters	0.264	gallons	gal
m^3	cubic meters	35.314	cubic feet	ft^3
m^3	cubic meters	1.307	cubic yards	yd^3
		MASS		
g	grams	0.035	ounces	oz
kg	kilograms	2.202	pounds	lb
Mg (or "t")	megagrams (or "metric ton")	1.103	short tons (2000 lb)	T
		TEMPERATURE (exact degrees)		
^oC	Celsius	1.8C+32	Fahrenheit	^oF
		ILLUMINATION		
lx	lux	0.0929	foot-candles	fc
cd/m^2	$candela/m^2$	0.2919	foot-Lamberts	fl
		FORCE and PRESSURE or STRESS		
N	newtons	0.225	poundforce	lbf
kPa	kilopascals	0.145	poundforce per square inch	lbf/in^2

*SI is the symbol for the International System of Units. Appropriate rounding should be made to comply with Section 4 of ASTM E380.
(Revised March 2003)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 BACKGROUND

There is a significant need to develop innovative technologies to detect pedestrians or other vulnerable road users at locations where they are experiencing increased exposure to dangerous traffic. The motivation of this research was triggered by the Fatality Analysis Reporting System data that indicated that there were 4,882 pedestrian fatalities and 64,000 pedestrian injuries in the United States in 2005.[(1)] Pedestrian-related traffic crashes accounted for 11 percent of the total fatalities and 2 percent of the total injuries in 2005. Europe and Japan have adopted standards to improve pedestrian protection, including passive energy absorption structure modification, active energy absorption systems installed on selected vehicle models from Jaguar and Citroën, and brake assist systems to reduce impact severity.

In 2007, there was a proposal to develop a real-time in-vehicle vision-based system that would detect pedestrians from a moving vehicle and estimate their position and distance relative to the vehicle at distances that would allow actionable warning time.

In April 2008, there was a panel meeting in Princeton, NJ, for the research team to demonstrate the technical capability in machine vision technology. The meeting also allowed the technical panel members to share the latest pedestrian fatality and injury statistics and collectively define the specifications of the desired research product.

According to information provided by the Federal Highway Administration (FHWA) technical panel, the following pedestrian fatalities occurred in 2007:[(2)]

- 73 percent occurred in an urban area.
- 76 percent occurred at non-intersection areas.
- 90 percent occurred under normal weather conditions.
- 67 percent occurred at night.

2007 pedestrian deaths by road type are as follows:

- 16 percent occurred on interstates or freeways.
- 55 percent occurred on other major roads.
- 27 percent occurred on minor roads.

2007 pedestrian deaths by speed limit in an urban environment are as follows:

- 24 percent occurred on roads with a speed limit less than 35 mi/h (56.35 km/h).

- 31 percent occurred on roads with a speed limit between 35 and 40 mi/h (56.35 and 64.4 km/h).
- 20 percent occurred on roads with a speed limit between 45 and 50 mi/h (72.45 and 80.5 km/h).
- 7 percent occurred on roads with a speed limit of 55 mi/h (88.55 km/h) or greater.
- 7 percent occurred on roads with an unknown speed limit.

The panel members also described scenarios such as pedestrian crossing patterns at intersections, pedestrians unexpectedly darting across the street from behind motor vehicles at midblock locations, etc. The performance requirements of the pedestrian detection system were formed based on the above inputs and the technology limitations known to the research team.

While several researchers have developed pedestrian detection systems, most of these systems suffer from a high false positive (FP) rate of detection (e.g., objects incorrectly detected as pedestrians). The proposed approach in this report attempts to alleviate this problem by following a layered approach (i.e., different layers of processing to gradually reduce FPs by using multiple cues (shape, appearance, and depth)) and classifying objects.

1.2 STUDY OBJECTIVES

The objectives of this study were to develop an in-vehicle pedestrian detection system capable of simultaneously recognizing pedestrians and other roadside infrastructures such as lamp posts, traffic signs, lane markings, pavements, buildings etc. The detection system should also be capable of distinguishing pedestrians and motor vehicles in and out of danger and determining danger levels. The researchers wanted to produce a system with a high detection rate and a low FP rate, coupled with inherently low-cost sensing technology for potential widespread adoption. The project addresses the Highway Safety Focus Area of the Broad Agency Announcement.

1.3 RESEARCH GOALS

The primary goal of this project was to develop a real-time in-vehicle system that uses stereo vision and advanced computer vision techniques to detect pedestrians under typical driving conditions and meet the metrics in table 1.

Table 1. Pedestrian detection performance specifications.

Pedestrian Detection Rate (percent)	FP Rate	Range of Detection (meters)	Conditions
98 (in path)	0.00001 (in path)	40 (day)	Benign mostly urban scenes
90–93 (out of path)	0.003 (out of path)	25 (night)	

1 ft = 0.305 m

These metrics were measured on selected video sequences by an offline implementation of the developed system. The *true positive rate* is defined as the percentage of pedestrians detected compared to the number of actual pedestrians in every frame of the sequence. It was measured on collected video sequences that contained a significant number of pedestrians. The *FP rate* is defined as the number of nonpedestrians incorrectly identified as pedestrians per hour of driving. It was measured by executing the real-time system in the test vehicle while driving under typical U.S. highway and urban conditions, collecting images every time the system detected a pedestrian, and manually checking the images against the actual presence or absence of pedestrians.

1.4 SCOPE OF REPORT

This report describes the work performed during the course of this project, which was partially funded by FHWA. It assumes a basic understanding of linear algebra, probability theory, and prior exposure to computer vision. Section 2 of this report provides a brief review of computer vision and probability theory topics that were used as prior art for the development of algorithms and software for a stereo-based pedestrian recognition system. Sections 3 and 4 describe the developed system configuration and the key innovations. The technical approach used to solve the pedestrian recognition problem is described in section 5. Section 6 describes the experiments conducted, results from collected data, and a comparison between state-of-the-art approaches published in literature. Finally, section 7 provides suggestions for future work and conclusions from this research work.

2. REVIEW OF RELATED TECHNOLOGIES AND RELATED WORK

This section provides a brief review of the computer vision technologies that were used in this project, and it discusses recent work in vision-based pedestrian detection.

2.1 RELATED TECHNOLOGIES

The key vision technologies used as background technology in this project include the following:

- Stereo vision for scene depth estimation.
- Appearance representation using histogram of oriented gradient (HOG) features extracted from the image.
- Classifiers for pedestrian and other object recognition.
- Markov random fields (MRFs) for a principled probabilistic method for scene structure labeling.

2.1.1 Stereo Vision

Stereo vision is a process of triangulation that determines range from two images taken from two different positions. These two images are taken simultaneously from a pair of cameras with a known baseline (i.e., separation distance between the cameras). In this implementation, the research team designed the camera setup so that the optical axes of the two cameras were almost parallel. The objective of a stereo vision system is to find correspondences in the images captured by the two cameras. This is usually done by some manner of image correlation and peak finding. The image correlation function is a local correlation-based method that provides a dense disparity map of the image, which can then be converted to a range map. The correlation function implemented is the sum of absolute differences (SAD). The equation for SAD for each pixel in an image when computed over a 7 × 7 pixel local region is found in figure 1.

$$\mathrm{SAD}(x, y, s) = \sum_{i=-3, j=-3}^{i=3, j=3} |A(x + i, y + j) - B(x + i - s, y + j)|$$

Figure 1. Equation. SAD.

Where:

A = Left image of the stereo pair.
B = Right image of the stereo pair.
x and y = Image pixel locations.
s = Number of horizontal shifts that are searched to find an image correlation.

There are other functions that could be used for local image correlation, including the sum of squared differences and normalized correlation. After the correlation was computed, 32 horizontal shifts in this case, the minimum value was detected and interpolated for an accurate

disparity estimate. In this implementation, the SAD correlation was applied to multiple resolutions of the image pair, extending the search range by a factor of two for every coarser resolution image. The disparity estimates obtained at coarser resolutions are generally less prone to false matches that can occur in regions of low texture, but they are commensurately less accurate.

The computed disparity maps based on these methods are often noisy because the range of data depends on accurately correlating each point in the image to a corresponding point in the other image. To increase the reliability of the range data, the image can be prefiltered (with boxcar or Gaussian filters), and the summing window for SAD can be changed from 7 × 7 pixels to 13 × 7 pixels. Additionally, the researchers masked out potentially unreliable data by computing a local texture measure and comparing it to a threshold. Researchers also compared the disparity estimates between the right image referenced to the left and the left image referenced to the right, checking for consistency between the two results. This checking method masks inconsistent or ambiguous disparity data such as areas that are occluded by one of the two cameras. Disparity maps computed at multiple resolutions are combined before range (depth) is computed.

Given the horizontal coordinates of corresponding pixels, x_l and x_r, in the left and right image, the range, z, can be expressed as follows:

$$z = \frac{b\,f}{d}$$

Figure 2. Equation. The range estimate of an image pixel.

Where:

b = The stereo camera baseline.
f = The focal length of the camera in pixels.
d = The image disparity value.

2.1.2 Feature Extraction Using HOG

HOG is a method of encoding and matching image patches under varying image orientation and scale changes. It is defined as the HOG directions of image pixels within a rectangular sampling window on an image. The gradient direction of each pixel in an image can be computed by convolving it with the Sobel mask (or differential kernels) in the X and Y directions. The ratio of convolution along two directions gives the underlying image feature direction. The gradient direction of each pixel is then binned in nine directions covering 180 degrees. HOG is then computed by gathering the directions of pixels inside a sampling window and weighting each response by the edge strength. This results in an 8 × 1 vector that is then normalized to be bound in a [0, 1] range. Figure 3 shows an illustration of how a HOG is computed. The image location is shown in blue in the photo of the pedestrian.

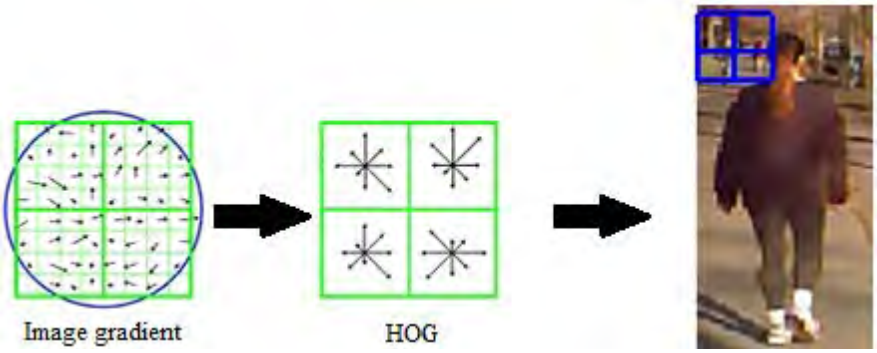


Figure 3. Illustration. HOG computed at an image location.

Within small regions, HOG encodes dominant shapes, which are computed by a voting scheme applied to the region's edge segments (see figure 4). Specifically, an image patch or region is first subdivided into multiple image cell regions. Each cell region is further divided into 2 × 2-pixel or 3 × 3-pixel local grids. The HOG feature is computed for each local grid region. For pedestrian recognition, candidate image patches are typically resized to a nominal size of 64 × 128 pixels, and HOG is computed. To handle image noise and exploit pedestrian shape, the algorithm applies a four-tap Gaussian filter to smooth the image and enhances it using histogram stretching.

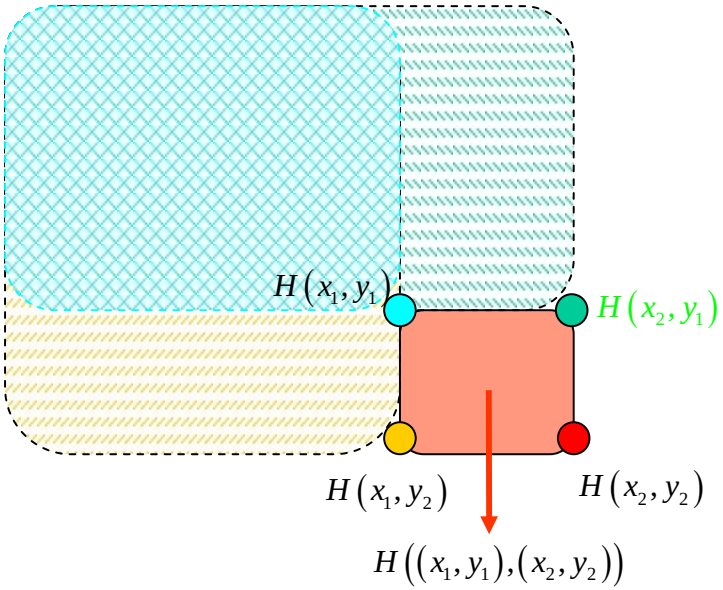


Figure 4. Illustration. HOG computation using integral images.

To efficiently compute HOG for use in a real-time system, an integral image is pre-computed so that HOG can be retrieved by look-up operations that consist of simple arithmetic summations. The integral image denotes a stack of image encoding where cumulative histogram of orientation for each pixel is computed by a fast scanning method. The integral histogram is computed as follows:

$$H(x_i, y_i) = H(x_i - 1, y_i) + H(x_i, y_i - 1) - H(x_i - 1, y_i - 1) + H(x_i, y_i)$$

Figure 5. Equation. Computation of integral histogram.

Given a candidate pedestrian region of interest (ROI), the corresponding HOG for each ROI is computed by sampling integral histogram as follows:

$$H\left((x_1, y_1), (x_2, y_2)\right) = H(x_2, y_2) - H(x_2, y_1) - H(x_1, y_2) + H(x_1, y_1)$$

Figure 6. Equation. Computation of HOG for a specific image patch.

2.1.3 AdaBoost Classifier

This project uses the AdaBoost classifier algorithm for multi-object recognition. The AdaBoost algorithm was introduced in 1995 by Freund and Schapire, and a tutorial is provided by Friedman et al.[3,4] Classifiers are supervised machine-learning procedures in which input test data are assigned to one of *N* labels based on a model that was learned from a representative training dataset. AdaBoost uses a training set $(x_1, y_1)\ldots\ldots\ldots(x_m, y_m)$ where x_i belongs to a domain *X* and y_i is a label in some label set *Y*. For simplicity, assume the labels are -1 or +1. AdaBoost calls a given weak learning algorithm repeatedly. It maintains a distribution or set of weights over the training set. Initially, all weights are set equally, but on each round, the weights of incorrectly classified examples are increased so that the weak learner is forced to focus on harder examples in the training set. The goodness of a weak hypothesis is measured by its error; this error is measured with respect to the distribution on which the weak learner was trained. The following information provides the pseudo-code for the algorithm:

Given $(x_1, y_1)\ldots\ldots\ldots(x_m, y_m)$ where $x_i \in X$, $y_i \in Y = \{-1, +1\}$:

Initialize $D_1(i) = {}^1/_m$

For $t = 1, \ldots\ldots, T$, use the following:

1. Train the weak learner using distribution D_t.
2. Get the weak hypothesis h_t: $X \rightarrow \{-1, +1\}$ with error $\epsilon_t = Pr_{i\sim D_t}\left[h_t(x_i) \neq y_i\right]$
3. Choose $\alpha_t = \frac{1}{2}\ln\left(\frac{1 - \epsilon_t}{\epsilon_t}\right)$

4. Update as follows:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases} = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x))}{Z_t}$$

   Where Z_t is a normalization factor.

5. Output the final hypothesis as follows:

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

2.1.4 MRF

One of the key contributions from this project was the development of a principled method to label scene structure into buildings, trees, other tall vertical structures (e.g., poles), and objects of interest (e.g., pedestrians and vehicles). This approach relied on posing the labeling problem as Bayesian labeling in which the solution is defined as the maximum a posteriori (MAP) probability estimate of the true labeling. This posterior is usually derived from a prior model and a likelihood model, which, in turn, depends on how prior constraints are expressed. The MRF theory encodes contextual constraints into the prior probability. MRF modeling can be performed in a systematic way as follows:

1. Pose the problem as one of labeling with a specific label configuration.

2. Further pose the problem as a Bayesian labeling problem in which the optimal solution is defined as the MAP label configuration.

3. Characterize the prior distribution of label configurations.

4. Determine the likely density of data based on an assumed observation model.

5. Use the Bayesian rule to derive the posterior distribution of label configurations.

A more detailed treatment of MRF models in computer vision is provided in *Markov Random Fields and Their Applications.*[5]

2.2 RELATED WORK

One of the most popular recent pedestrian detection algorithms is the HOG method created by Dalal and Triggs.[6] They characterized pedestrian regions in an image using HOG descriptors, which are a variant of the well-known scale invariant feature transform (SIFT) descriptor.[7] Unlike SIFT, which is sparse, the HOG descriptor offers a denser representation of an image region by tessellating it into cells which are further grouped into overlapping blocks. At each cell, a HOG at pixels belonging to the cell is computed. Within each block, a HOG descriptor is calculated by concatenating individual cell histograms belonging to that block and normalizing the resulting feature vector to give some degree of illumination invariance. A

two-class support vector machine (SVM) classifier was trained using the HOG features and used for final pedestrian detection.

Dalal and Triggs reported significantly better results compared to previous approaches based on wavelets and a principle component analysis SIFT of around 90 percent correct pedestrian detection at 10^{-4} FPs per number of windows (FPPW) evaluated.[6,8,9] Note that based on the image size and the number of scales used to detect pedestrians, a FP rate of 10^{-4} FPPW corresponds to about 0.4 FPs per frame (FPPF).

Tuzel et al. proposed the covariance descriptor to characterize global image regions and used a Riemannian manifold for pedestrian detection.[10] They reported improved results of about 93.2 percent correct detection compared to the HOG descriptor at the same rate of 10^{-4} FPPW. Tran and Forsyth used geometric features describing the spatial layout of parts with appearance features characterizing individual parts.[11] They employed structured learning to determine the discriminative configuration of parts and reported excellent detection rates exceeding 95 percent with 0.1 FPPF (10^{-4} FPPW naive Bayes weight) less number of window needed to be evaluated since the approach is robust to centering of the pedestrian ROI) on the *INRIA Person Dataset*, though no time performance was discussed.[12] Wu and Nevatia used a cluster boosted tree classifier for pedestrian detection and also showed a performance of 95 percent at 10^{-4} FPPW.[13]

Leibe et al. described a stereo-based system for three-dimensional (3D) dynamic scene analysis from a moving platform, which integrates a sparse 3D structure estimation with multicue image-based descriptors (shape context) computed using Harris-Laplace and HOG features to detect pedestrians.[14,15] The authors showed that the use of sparse 3D structure significantly improved the performance of pedestrian detection. The best performance cited was 40 percent probability of detection at 1.65 FPPF. While the structure estimation was performed in real time, the pedestrian detection was significantly slower.

Gavrila and Munder proposed Preventive Safety for Unprotected Road User (PROTECTOR), a real-time stereo system for pedestrian detection and tracking.[16] PROTECTOR employs sparse stereo and temporal consistency to increase the reliability and mitigate misses. Gavrila and Munder reported 71 percent pedestrian detection performance at 0.1 FPPF without using a temporal constraint with pedestrians located less than 82 ft (25 m) from the cameras. However, the datasets used were from relatively sparse, uncluttered environments. Recently, Doll´ar et al. introduced a new pedestrian dataset and benchmarked a number of existing approaches.[17]

Another leading real-time monocular vision system for pedestrian detection was proposed by Shashua et al.[18] A focus of attention mechanism was used to rapidly detect candidates. The window candidates (approximately 70 per frame) were classified as pedestrians or nonpedestrians using a two-stage classifier. Each input window was divided in 13 image subregions. At each region, a histogram of image gradients was computed and used to train an SVM classifier. The training data were divided into nine mutually exclusive clusters to account for pose changes in the human body. The 13×9 vector containing the response of the SVM classifiers for each of the nine training clusters were used to train, as well as AdaBoost second-stage classifier. A practical pedestrian awareness system requires few FPs per hour of driving. As a result, the authors employed temporal information to improve the per-frame pedestrian

detection performance and to separate in-path and out-of-path pedestrian detections, which increased the latent period in the system.

Hoeim et al. presented a method for learning 3D context from a single image by using appearance cues to infer simple geometric labelings.[19] Hoiem et al. also presented a probabilistic detection framework which exploits the overall 3D context extracted using *Geometric Context from a Single Image*.[19] The authors argued that object recognition could not be solved locally but required statistical reasoning over the whole image.[20]

Wojek and Schiele proposed a probabilistically sound combination of scene labeling and object detection using a conditional random field, but their method relied on appearance rather than 3D.[21] Brostow et al. investigated the use of 3D features from structure-from-motion to classify patches in the scene.[22]

3. SYSTEM CONFIGURATION

The developed system consists of the following components:

- National Television System Committee (NTSC) cameras in a parallel stereo configuration with a baseline of 7 inches (177.8 mm). Each camera has a 46-degree horizontal field of view (see figure 7 and figure 8).
- Acadia I™ Vision Accelerator Board for real-time stereo depth estimation (see figure 9).
- Dual Intel Core2Quad processor running Windows XP® or later.
- Monitor and keyboard installed in the motor vehicle.

Figure 7. Photo. In-vehicle camera sensor and a portable briefcase processing system.

Figure 8. Photo. NTSC camera used in the developed pedestrian detection system.

Figure 9. Photo. Acadia I™ vision accelerator board.

The research team used a modified Lincoln® Navigator and a Toyota® Highlander to test the developed real-time system. Both vehicles were modified to include an inverter that was used to supply power to the components listed above.

4. KEY INNOVATIONS

The research team made the following key innovations during the execution of this project:

- Detected and recognized multiple roadway objects in real-time. In addition to pedestrian detection, the system also detected regions that corresponded to buildings, poles, trees, and motor vehicles.

- Used multiple cues to detect, classify, and track people. Stereo images provided depth information. Shape and appearance were used in the classifier, and motion cues were used to perform object tracking.

- Used contextual information about regions such as buildings, trees, poles, etc., derived from depth information to reject a majority of the FPs that were detected by the initial pedestrian detection module.

5. TECHNICAL APPROACH

This section describes the technical approach to achieve the research goals. The research team captured data from a calibrated stereo rig mounted behind the rear-view mirror of a car. The data was processed at 30 frames per second using an Acadia I™ Vision Accelerator Board to compute dense disparity maps at multiple resolution scales using a pyramid image representation and a SAD-based stereo matching algorithm.[23,24] The disparities are generated at three different pyramid resolutions, D_i, i = 1, 2, …3, with D_0 being the resolution of the input image. In figure 10, the pedestrian detector (PD) module takes the individual disparity maps and converts each one into a depth representation. These three depth images are used separately to detect pedestrians using a template matching of a 3D human shape model, as described in detail in section 5.2 of this report. The structure classifier (SC) module employs a combined depth map to classify image regions into several broad categories such as tall vertical structures, overhanging structures, and ground and poles to remove pedestrian candidate regions that have a significant overlap. Finally, the pedestrian classifier (PC) module takes the list of pedestrian ROIs provided from stereo modules and confirms valid detections by using a cascade of classifiers tuned for several depth bands and trained on a combination of pedestrian contour and gradient features. The rest of this section describes the algorithms implemented and the results produced by each stage.

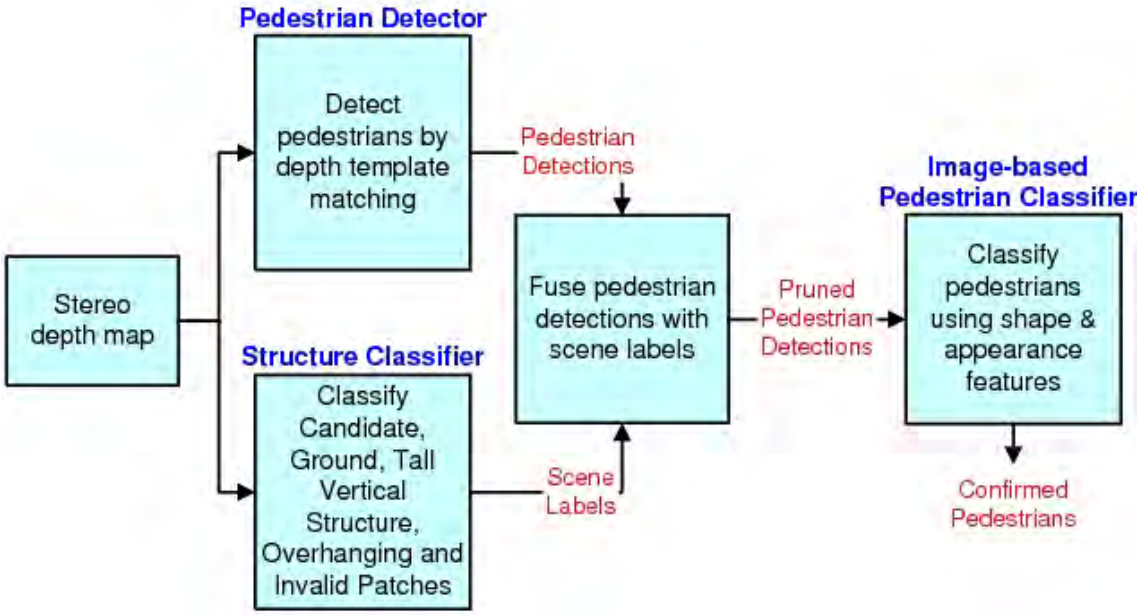


Figure 10. Illustration. Diagram of the developed system.

5.1 SENSORS

The proposed system consists of a stereo rig that is made of off-the-shelf monochrome cameras and the Acadia I™ Vision Accelerator Board. The cameras are standard NTSC format 720 × 480 resolution with a 46-degree horizontal field of view.

5.2 STEREO-BASED PEDESTRIAN DETECTION

The approach to stereo-based generic object detection framework is based on the techniques introduced by Chang et al.[25] The algorithm introduced by Chang et al. used template matching (through correlation) of pre-rendered 3D templates of objects (e.g., pedestrians and motor vehicles) with the depth map to detect objects.[25] The 3D template matching was conducted in a coarse to fine manner over a two-dimensional (2D) grid overlayed onto the local XY plane. At each grid location, a 3D template was matched to the range image data by searching around the X, Y, and Z directions according to the local pitch uncertainty due to calibrations and bumps in the road surface. Locations on the horizontal grid corresponding to local maximal correlation were returned as candidate object locations.

In the proposed method, template matching is conducted separately using a 3D pedestrian shape template in three disjoint range bands in front of the host vehicle. The 3D shape size is a determined function of the actual range from the cameras. The researchers obtains depth maps at separate image resolutions, D_i, $i = 1, 2, \ldots 3$. For the closest range band, the researchers employed the coarsest depth map D_2, for the next band level D_1, and for the furthest band the finest depth map D_0. This ensures that at each location on the horizontal grid, only the highest resolution disparity map that is dense enough is used. The output of this template matching is a correlation score map (over the horizontal 2D grid) from which peaks are selected by non-maximal suppression as in Chang et al.[25]

Note that this detection stage must ensure small pedestrian miss rates. As a result, a larger number of peaks obtained by non-maximal suppression is acceptable. The researchers relied on additional steps to reduce the candidates. Around each peak, the area of the correlation score map with values within 60 percent of the peak score was projected into the image to obtain the initial pedestrian ROI candidate set. This set was further pruned by considering the overlap between multiple ROIs: detections with more than 70 percent overlap with existing detections were removed. After this pruning step, a Canny edge map was computed for each initial pedestrian ROI. The edge pixels that were too far off from the expected disparity were rejected. A vertical projection of the remaining edges resulted in a one-dimensional profile from which peaks were detected using mean shift.[26] A new pedestrian ROI was initialized at each detected peak, which was refined first in the horizontal direction followed by the vertical direction to get a more centered and tightly fitting bounding box on the pedestrian. This involves using vertical and horizontal projections of binary disparity maps (similar to using the edge pixels above) followed by detection of peak and valley locations in the computed projections. After this refinement, any resulting overlapping detections were again removed from the detection list. The above approach allows detections of pedestrians and vehicles up to a range of 131.2 ft (40 m). Figure 11 through figure 14 show examples of pedestrian detection performance. In the figures, the white boxes indicate possible pedestrians, and the blue boxes indicate possible pedestrians to be further analyzed by an appearance classifier. Both true detections and typical FPs are shown. The objective of the following modules is to reduce the FPs.

5.3 EXAMPLES OF PEDESTRIAN DETECTION

©INRIA (See Acknowledgements section)

Figure 11. Photo. Example 1 of pedestrian detection.[(12)]

©INRIA (See Acknowledgements section)

Figure 12. Photo. Example 2 of pedestrian detection.[(12)]

©INRIA (See Acknowledgements section)

Figure 13. Photo. Example 3 of pedestrian detection.[(12)]

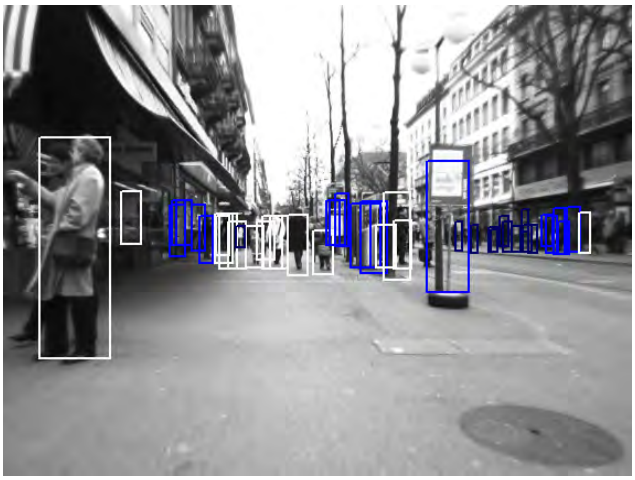


Figure 14. Photo. Example 4 of pedestrian detection.[12]

5.3.1 Structure Classifier

A key step in the developed method for pedestrian detection is depth-based classification of the scene into a few major structural components. Given an image and a sparse and noisy range map, the goal is to probabilistically label each pixel as belonging to one of the following scene classes:

- **V**: Tall vertical structure (magenta).
- **O**: Overhanging structure (green).
- **G**: Ground (yellow).
- **C**: Candidate objects (blue).

An occupied cell in the range map of a scene provides evidence for the presence of one or more of the structure classes. The structure classes outlined above typically span multiple adjacent cells in a scene with discontinuities at the boundaries of the classes. Therefore, local evidence for the presence/absence of a class can be combined with neighborhood constraints to probabilistically estimate the class labels.

The range map from the stereo does not provide enough resolution to differentiate between a group of people and a motor vehicle. As a result, the research team labels all motor vehicle-like objects as object candidates and allows the appearance-based classifier to resolve detections in these regions. These classes have been chosen to competitively label pixels among a few commonly occurring structures as a precursor to PC versus non-PC. This is in contrast to traditional detectors that directly apply PC/non-PC in which the negative examples themselves form a large set of structured classes. The research team further separates the structured classes into classes that are distinct from the pedestrian class. In this method, if large numbers of pixels can be rejected as being part of generic structural classes, the system substantially reduces the number of false hypotheses that are presented to a PC/non-PC, gaining both in performance (FP rate) and computation.

The research team performed structure classification using depth maps. An example depth map is shown in figure 15. The map is pseudo-colored with red denoting close-range objects, cyan denoting far-off objects, and black denoting missing depth. The depth map illustrates a number of issues: (1) objects appear bloated in the range map due to the stereo integration window, (2) the characteristic noise in the range values is observable as scattered fragments, and (3) the occlusion boundaries between objects are noisy.

To handle depth map errors, first, the research team defines a structure called the vertical support histogram (VSH) to accumulate 3D information over voxels in the vertical direction (see figure 15). In a given frame, the system will compute a feature vector using this structure and subsequently use the feature vector to learn the likelihood of each pixel belonging to a given structural class. Next, the team makes use of the scene-context constraints arising from the camera viewpoint by formulating the labeling problem as an MRF, where the smoothness constraints allows the team to reason about the relative positioning of the 3D structure labels in the image. This reduces error in labeling due to depth inaccuracies and gives a smooth labeling of the scene.

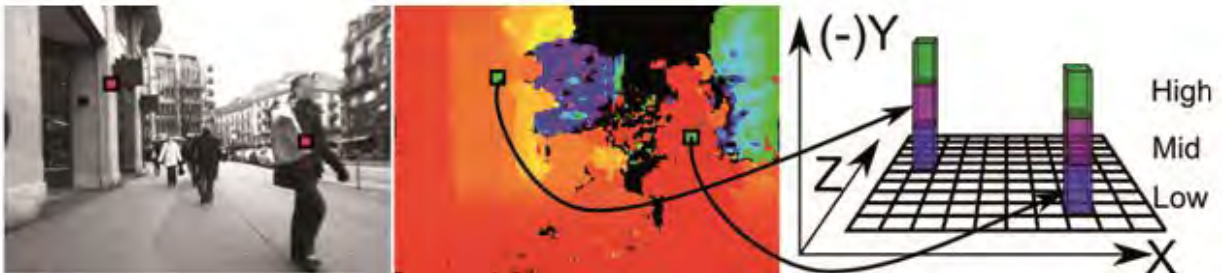


Figure 15. Illustration. VSH.[12]

5.3.2 Bayesian Labeling

The main problem with using Bayesian labeling is deriving a labeling $L = \mathrm{l}$ of image patches, Π, using a set of image observations, r. Suppose that both the a priori probabilities $P(\ell)$ of labels ℓ and the likelihood densities $p(\ell \mid r)$ of r are known, the best estimate one can get from these is one that maximizes MAP, which can be computed using the Bayesian rule as follows:

$$P(\ell \mid r) = \frac{p(r \mid \ell)P(\ell)}{p(r)}$$

Figure 16. Equation. Bayesian rule.

In the above equation, $p(r)$, which is the density function of r, does not affect the MAP solution.

The following section describes the approach to estimate the likelihood densities $p(r \mid \ell)$ and the prior probabilities $P(\ell)$ for this labeling problem.

5.3.3 Likelihood Densities of Structure Labels

The likelihood densities for the structure labels are estimated by first computing VSH, determining the likelihood of the structure labels using VSH information, and modeling the smoothness inherent in scene structures. Each of these steps is described in more detail below.

The 3D scene is represented as distributions of reconstructed 3D points with respect to a ground plane coordinate system. The ground plane can be estimated using several well-known techniques applied to the reconstructed stereo points, such as in Leibe et al.[14] The ground plane (XZ in this case) is divided into a regular grid at a resolution of $X_{res} \times Z_{res}$. At each grid cell, a histogram of distribution is created of 3D points according to their heights. All the image pixels that map into a given XZ coordinate participate in that cell's histogram. The heights, or Y coordinates, of all the points in a cell are mapped into a *k*-bin histogram where each bin represents a vertical height range. This structure is named VSH and is denoted by *V*. At any given grid cell, the following equation can be used:

$$V[g(X,Z)] = [s_1^g, s_2^g, ..., s_k^g]^T$$

Figure 17. Equation. VSH for a given grid cell.

In this equation, s_i^g measures the support for the *i*th-bin of the histogram. Figure 8 shows how image points and the corresponding depth estimates are mapped to 3D distributions for an example histogram with $k = 3$ bins.

Three ranges are chosen to capture the typical vertical characteristics of structures of interest which result in three histograms: h_{low}, h_{mid}, and h_{hi}.

In order to compute the supports, s^g, from noisy range estimates at each pixel, the researchers uses a mean-around-the-median estimate of range. If a *wxh* patch is defined at each pixel (*X*, *Y*), a robust range estimate is computed for each patch (in the following, pixel and patch are used interchangeably, with the idea that the context makes the sense clear). Image points, (*X*, *Y*), with the range estimate, *Z*, are mapped to the corresponding (*X*, *Z*) grid cell with height estimate *Y*. *Y* is used to increment the appropriate bin of VSH at (*X*, *Z*).

Each cell of the histogram is normalized by dividing with the maximum number of pixels that can project to the cell. For a cell at a distance Z from the camera (with horizontal and vertical focal-lengths f_x and f_y, respectively), the maximum number of pixels in each image row is as follows:

$$N_{row}^{\max}(Z) = X_{res} \bullet \frac{f_x}{Z}$$

Figure 18. Equation. Maximum number of pixels in each image row.

The maximum number of image rows in the height-band (H_{min} and H_{max}) is as follows:

$$N_{col}^{\max}(Z) = (H_{\max} - H_{\min}) \bullet \frac{f_y}{Z}$$

Figure 19. Equation. Maximum number of image rows in a specific height band.

In this equation, $H_{\max}$ is determined taking into account the maximum height that is visible in the image at distance Z . This gives the normalizing factor for the cell as follows:

$$N(Z) = N_{row}^{\max}(Z) \bullet N_{col}^{\max}(Z)$$

Figure 20. Equation. Normalization factor for each cell in which VSH is calculated.

V(X, Z) is defined in 3D space. If this 3D representation is transferred to the 2D image and augmented with the 3D height, then, at a given image patch, *p*, the robust range estimate *Z* can be used to project this patch to a footprint (collection of cells) in the XZ-grid coordinate system. An aggregate of the VSH values for the cells within this footprint serves as the total support of p. H^P is defined as the average height estimate of the image pixels within the patch. Subsequently, each such *p* is associated with a *k* + 1 - *D* feature vector as follows:

$$r_p = [V(X,Z)^T, H^p]^T = [s^{p_1}, s^{p_2}, ...s^{p_k}, H^p]^T$$

Figure 21. Equation. Feature vector extracted from each image patch.

VSH captures the distribution of 3D points in any given scene in terms of quantized height bins. *V*(*X*, *Z*) is a representation of the scene in front of a camera. In order to associate each image patch with structural labels, the researchers compute the likelihoods for the augmented feature vector, r_p, conditioned on the specific structural labels defined earlier.

The research team randomly sampled approximately 100 frames from sequences in typical urban driving scenarios. In each frame, structures were coarsely hand-labeled as tall vertical structures (buildings), candidate objects (pedestrians, vehicles, etc.), ground, and overhanging structures. The research team experimented with the number of histogram bins and the placement of the bin boundaries and empirically derived the three most discriminative feature components (bins in this case). Feature vectors along these three most discriminative components for all the labeled patches r_p are shown in figure 22, with different colors denoting different ground truth labels. The bin boundary values for these bins are in table 2. The resolution was 12 × 16 pixels. This separation is not surprising and can be explained as follows:

- All buildings should at least have support in h_{mid}.
- All candidate objects should have a low H_p and at least have support from h_{low} (and some support from h_{hi} when under overhanging structures) and all overhanging structures should have a high H_p and at least have support from h_{hi} and lack of support from h_{mid}.

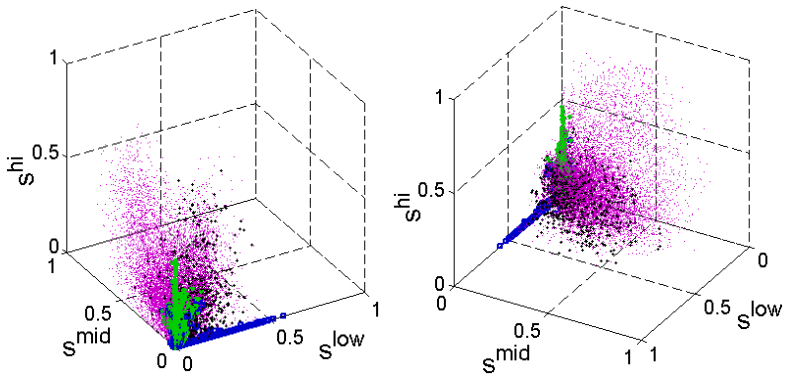


Figure 22. Illustration. Two views of the feature space showing the distribution of vectors from which the class conditional likelihoods are estimated.

Table 2. Parameter settings for SC.

XY Histogram (meters)					MRF	
X_{res}	Z_{res}	h_{low}	h_{mid}	h_{hi}	Z_n	ρ_{bp}
0.1	0.1	0 to 2	2 to 4	4 to 8	0.1	1.0

1 ft = 0.305 m

Figure 23 through figure 26 show the various steps of the likelihood density estimation process for one frame. Note that, in particular, the vertical structure likelihoods in figure 26 capture the visible extent of the buildings all the way to the base, a task that is difficult to achieve with a simple heuristic on *H*.

Figure 23. Photo. Likelihood density estimation of original (left) and labeled structures (right) showing buildings and candidate objects.[(12)]

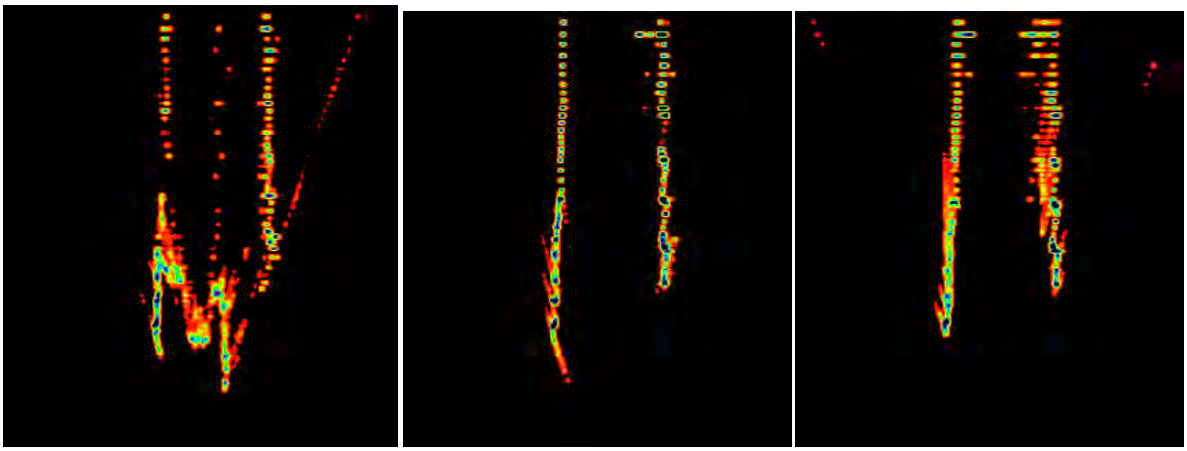


Figure 24. Illustration. Top view of VSH components: h_{low} (left), h_{mid} (center), and h_{hi} (right).

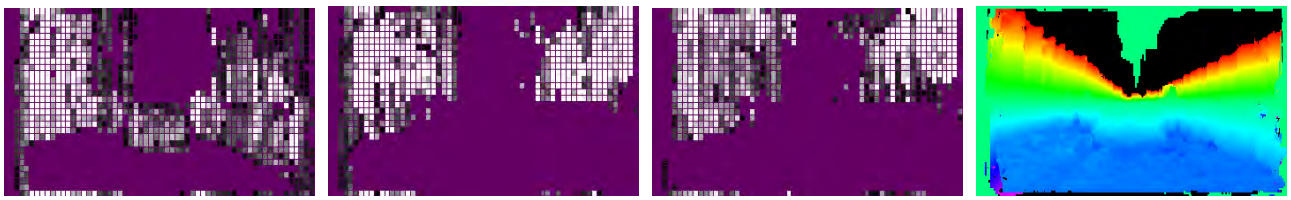


Figure 25. Illustration. VSH projected on to the image (left three images) and the height of each pixel (right).

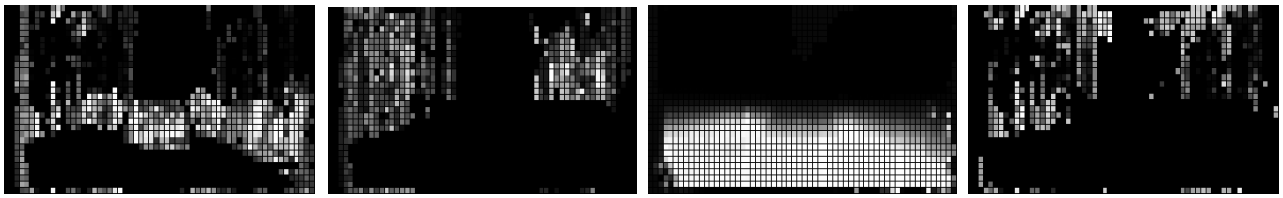


Figure 26. Illustration. Likelihoods conditioned on the four labels: candidate objects, vertical structures, ground, and overhanging structures.

The research team performed kernel density estimation on the feature space obtained in the above process to compute the likelihood densities, $p(r \mid \ell)$, for each of the four class labels, ℓ, as follows:[27]

$$p(r \mid \ell) = \frac{1}{n^c} \sum_{i=1}^{n^c} K_H (r - r_i^c)$$

Figure 27. Equation. Kernel density estimation on the feature vector extracted from each image patch.

Where:

r_i^c = The feature vectors of all the patches.
i = The training set belonging to the class c.
$K_H(u) = \alpha(H)K(H^{\frac{-1}{2}}u)$ = A kernel function.
H = A bandwidth matrix which scales the kernel support to be radially symmetric.

In implementation, the research team defines $K(u) = k(u^T u)$ and uses the following bi-weight kernel:

$$k(u) = \left\{ \frac{(1-u)^3 .....0 \le u \le 1}{0.....u > 1} \right\}$$

Figure 28. Equation. Bi-weight kernel used in the kernel density estimation function.

The bi-weight kernel is efficient to compute, and the research team found that it had a comparable performance to more complex kernels.

In addition to the likelihoods of structural labels, the research team modeled the smoothness inherent in scene structures through MRF priors on a pair-wise basis. The a priori joint probability of labels, $P(L = \ell)$, is difficult to define in general but is tractable for MRFs. If L is represented as an MRF, then the prior probability, $P(\ell)$, is a Gibbs distribution given by the following equation:[28]

$$P(\ell) \propto e^{-E_s(\ell)}$$

Figure 29. Equation. Gibbs distribution used to model the prior probability.

In the above equation, $E_s(\ell)$ is the cost associated with ℓ and is modeled as a pair-wise smoothness term between neighboring patches. L can be formulated as an MRF on the grid graph represented by the patch grid, P_i, with the four-connectivity imposed by the grid structure defining the edges. This can occur if the following conditions are satisfied:

- L is a random field.
- The label for a particular patch given those of all other patches depends only on the labels of the neighboring patches.

These are reasonable assumptions in this scenario. For example, the identification of a patch as a building patch might depend on whether its neighboring patches are ground but has little to do with the identity of the patches far away spatially.

The next step is to define the smoothness cost, E_s, from which the prior probability $P(L = \ell)$ can be computed. The smoothness term can be used to model valid configurations of scene objects possible from the camera viewpoint. Thus, for each patch, its neighboring patch will be considered, and the cost of associating a pair of labels with the two patches will be defined. The neighboring patch is defined as the patch that is four-connected to this patch and is also close in its world depth, Z^w. Thus, two patches, which are neighbors in the image space but

distant in the world space, are treated to have no conditional dependence on each other's labeling in the MRF network. This condition essentially cuts the grid graph along depth discontinuities before the MRF framework starts any label propagation. The remaining neighbors are now depth neighbors as well, and it is easier to reason about what objects can (or cannot) be near other objects.

Let p and q be two neighboring patches from the patch grid and Z_p^w and Z_q^w represent the world depths of these patches. Define a binary variable, $\rho_n = \partial(t)$, as follows:

$$\partial(t) = \begin{cases} 1 \text{.....} \dfrac{|Z_p^w - Z_q^w| < Z_n}{Z_p^w} \\ 0 \text{.....} otherwise \end{cases}$$

Figure 30. Equation. Binary variable used to test the depth neighborhood of image patches.

The binary variable defined in figure 30 is used for testing depth neighborhood using a ratio threshold, Z_n. The smoothness cost assigned to the patch pair (p, q) is as follows:

$$E_s = \rho_{bp}\, \rho_n\, D(p, q, L(p), L(q))$$

Figure 31. Equation. Smoothness cost associated with each image patch pair.

In the above equation, ρ_{bp} is the constant weight factor applied to the smoothness term and is set empirically, $L(p)$ and $L(q)$ are the labels of p and q, and $D(.)$ is a function that measures the compatibility between those labels.

The function $D(.)$ is defined by considering not only the labels $L(p)$ and $L(q)$, but also considering if patch p is a left, right, top, or bottom neighbor of patch q. The function can enforce different costs for the same pair of labels ($L(p)$ and $L(q)$) if p is below q compared to if p is above q. For example, if p is a building patch and q is below p, then q can be a building, candidate object, or ground patch. However, if q is above p, then q can only be a building patch since one cannot expect to see either ground or candidate objects along the top edge of a building. Note that in the first scenario, the candidate label is included to allow a pedestrian patch close in depth to the building patch to occlude the lower part of the building. The allowed choices for $L(q)$ would also be the same as the first scenario if p and q were horizontal neighbors. $D(.)$ is a binary function which imposes a penalty 1 (correction was applied) if a pair of labels is inconsistent and a penalty 0 (no correction) otherwise. In implementation, the MAP estimation (see figure 16) is done with the max-product belief propagation algorithm.[29]

5.4 PEDESTRIAN CLASSIFIER

The PC layer consists of a set of multirange classifiers. Specifically, three classifiers are trained for distance intervals of 0 to 65.6, 65.6 to 98.4, and 98.4 to 131.2 ft (0 to 20, 20 to 30, and 30 to 40 m) where a specific layer is triggered based on distance of detected target.

This is inspired by the fact that under a typical interlaced automotive grade camera with a resolution of 720 × 480 pixels, pedestrian ROI size on image varies substantially. For example, people who are 98.4 ft (30 m) away or farther appear on image around 25 pixels or smaller. Thus, it is desirable to handle them with approaches tuned to each specific resolution variations rather than from a single classifier covering mixed resolutions.

Each of the three distance-specific classifiers is composed of multiple cascade layers to efficiently remove FPs. For the optimal performance of the target application, the classifiers are designed with different approaches (i.e., for low latency detection at short ranges and detection at farther distances).

5.4.1 Contour-Based Classifier

The first classifier is designed to reliably classify high-resolution pedestrians in a computationally efficient manner. In general, for pedestrian detection approaches reported thus far, it is often required to search for optimal ROI position and size to obtain valid classification scores.[6,10,14] This is due to the sensitivity of classifier to ROI alignments that results from rigid local feature sub-ROI placement inside the detection window.

This result would require exhaustive search over multiple positions and scales for each input ROI. Aside from computational overhead, the classification score also becomes sensitive and often produces false negatives.

Note that there are approaches, such as codebook-based approaches, that do not require global ROI search; however, they typically show inferior performance to approaches with fixed sub-ROI.[14,6] False negatives can also happen when pedestrians appear against a complex background (i.e., highly textured). In this case, typical image gradient-based features become fragile due to the presence of multiple gradient directions in a local image patch.

The research team addressed this issue by designing a classifier that combines contour template and HOG descriptors, which helps with local parts alignment and background filtering (see figure 32).

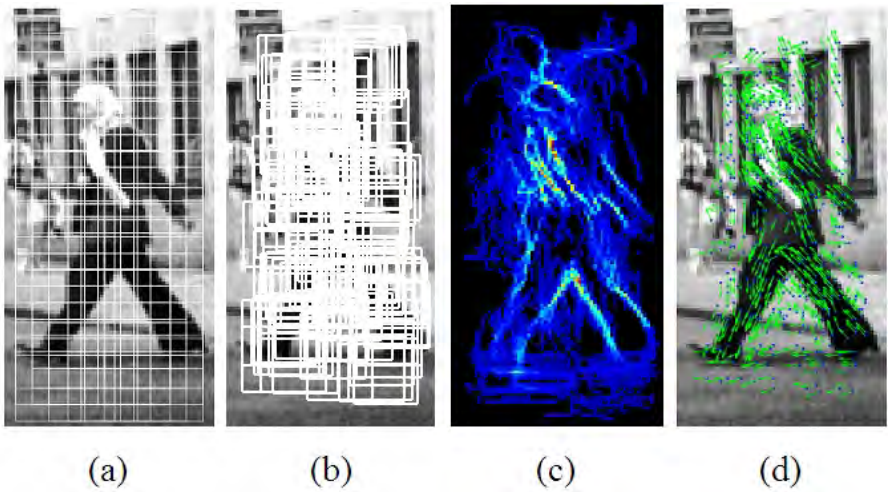


Figure 32. Photo. Process of contour and HOG classification for (a) fixed sub-ROI, (b) local ROI, (c) foreground mask from contour matching, and (d) filtered HOG directions underlying masked regions.

The research team uses a collection of templates of shape contours for each local feature window. That is, each feature window (i.e., sub-ROI) contains examples of contour models of underlying body parts that can cover different variations. For example, a sub-ROI at head position contains a set of head contours samples of different poses and shapes. Figure 33 provides an example of local contour models.

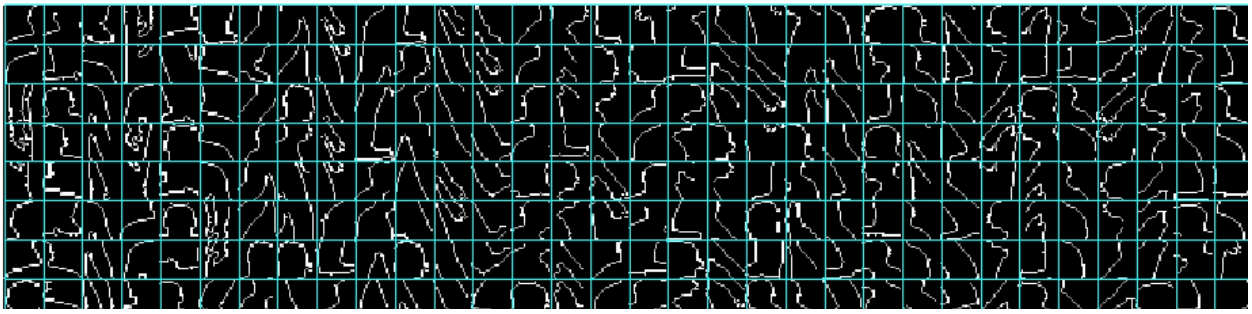


Figure 33. Illustration. Example of local contour models.

Given a pedestrian ROI, each local feature window can search in a limited range and lock on underlying local body parts. In addition to computational efficiency, it can better handle local parts deformation from pose changes and shape variations. It can also overcome ROI alignment issues.

The part contour models consist of edge maps of representative examples. Each sub-ROI contains 5 to 12 such templates. Contour template matching is achieved by chamfer matching. For each sub-ROI, the chamfer score is computed for each template model. The refined sub-ROI position is then obtained from mean position of maximum chamfer scores from each template (see figure 34).

From the contour template, a foreground mask can also be composed by overlapping binary local templates at each detected position that is weighted by matching scores. The foreground mask is used as a filter to suppress noisy background features prior to classification step.

$$Ctr_{subROI}(i_{x,y}) = \alpha \sum_{i \in voc(ixy)} w_{ch}(i) Ctr_{templ}(i; I_{ch})$$

$$M^{FG}(i_{x,y}) = M^{FG}(i_{x,y}) + \alpha \sum_{i \in voc(ixy)} w_{ch}(i) I^{Cont}_{templ}(i)$$

Figure 34. Equation. Foreground mask for the contour template.

In the above equation, $\alpha = \frac{1}{\sum_i w_{ch}(i)}$

Where:

$Ctr_{subROI}(i_{x,y})$ = Center of local sub-ROI.
M^{FG} = Foreground mask.
I^{Cont}_{templ} = Binary contour template.
$Ctr_{templ}(i; I_{ch})$ = Center from chamfer matching score with the *i*th kernel image.

Given refined sub-ROI and foreground mask, the research team applied a HOG-based classifier. The HOG feature is computed by using refined sub-ROI boxes where gradient values are enhanced by the weighted foreground mask.

Each of the three images displays the original image, the foreground mask generated from local part templates, and the resulting edge filtering. Note that local contour parts can capture global body contours at various poses from its combinations; however, this does not form a conforming pedestrian mask for negative patches.

Figure 35 shows examples of a foreground mask and negative patches on pedestrians. Columns 3, 6, and 9 in figure 35 show the results on negative data. On the pedestrian images, the proposed scheme can refine the ROI positions on top of matching local body parts and can enhance effectively underlying body contours. The mask also produces nonconforming shape and position on negative examples. This scheme produces efficient and reliable performance on relatively high-resolution pedestrian ROIs. However, as pedestrian ROI size becomes smaller, it faces a problem as contour extraction and matching steps become fragile under low-resolution images. As a result, the researchers employ conventional HOG classifier at farther distances of pedestrian ROI (less than 35 vertical pixels).

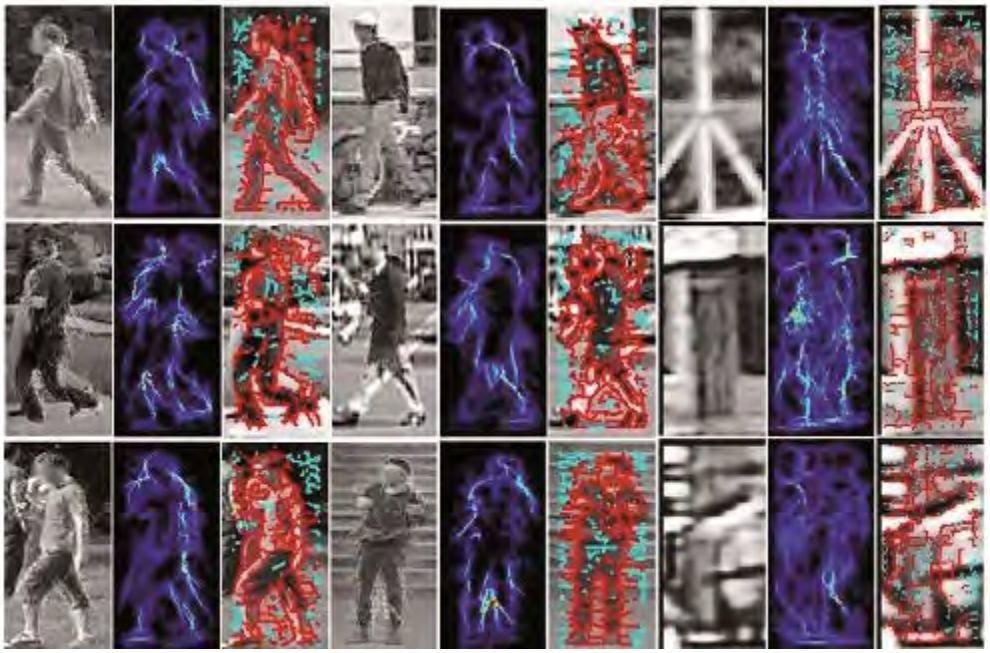


Figure 35. Photo. Foreground mask examples.

5.4.2 Far Distance Classification

At the second and third levels, the research team uses a cascade of HOG-based classifiers. The HOG classifier proved to be effective on relatively low-resolution imageries when body contour is distinctive from the background.

Each classifier is trained separately for each resolution band. Gaussian smoothing and subsampling is applied to match target image resolution, where 25 (at 82 ft (25 m)) and 17 (at (114.8 ft (35 m)) are nominal pixel heights for the distance interval.

Note that at farther distances, the image contrast is reduced as pedestrian ROI size becomes smaller. To compensate for this reduction and to meet scene dependent low-light situations, a histogram normalization step is used that is based on histogram stretching. For each ROI, the research team applies local histogram stretching where the top 95 percent gray value histogram range is linearly extended to cover 255 gray levels. As opposed to histogram normalization, it does not produce artifacts at low contract images, yet it can enhance underlying contours.

5.5 TRACKING

The research team implemented a pedestrian tracking method designed to complement intermittent missing target detection and to allow further analysis of spatial-temporal feature spaces (i.e., motion cues) to enhance classification performance. Figure 36 provides an overview of the pedestrian tracker. The red box indicates the object inside it was identified as a pedestrian, and the green shading indicates the pixel region (pedestrian's trace) that was occupied by the pedestrian in subsequent frames.

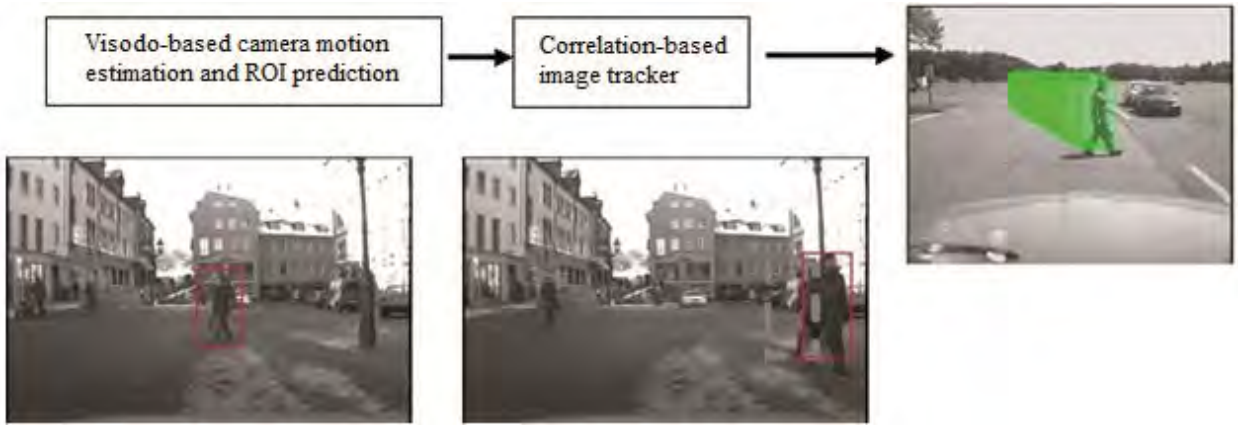


Figure 36. Photo. Overview of the pedestrian tracker.

5.5.1 Camera Motion Estimation

The tracking method consists of two steps: (1) 3D feature-based camera motion estimation and (2) image correlation-based ROI tracking. The research team's visual odometry system computes 3D motion of the camera, specifically rotation and translation of a vehicle between adjacent frames with respect to the ground plane. To compute camera motion, the system first extracts feature points on each frame where features are obtained from corner points from various scene structures. Next, the correspondences between adjacent frames are established by using random sample consensus-based point association. Given correspondences, the relative camera motion can be computed by solving the structure from motion equation.

5.5.2 Image Correlation-Based Tracker

The estimated camera motion parameter is used to predict the location of the detected pedestrian boxes on image (ROI) in the current frame. The camera motion-based prediction is important to accurately localize ROIs under large image motions such as turning.

Given the predicted location of the ROI from the previous frame (*t*-1), the new location in the current frame (*t*) is estimated by patch correlation-based tracker module. The correlation-based tracker refines ROI position by searching through multiple candidate positions and scales of the enlarged prediction window that matches the highest appearance similarity with the corresponding ROI image patch. Figure 37 provides an equation for an image correlation tracker.

$$I_{new}^{t}(x,y) = \arg\min_{x,y}\left(MNCC\left(I_{patch}^{t-1}, I_{search}^{t}\right)\right)$$

$$\text{where } MNCC\left(I_j, I_k\right) = \frac{\sum_{x,y}\left(I_j - m\left(I_j\right)\right)\left(I_k - m\left(I_k\right)\right)}{\text{var}\left(I_j\right)\text{var}\left(I_k\right)}$$

Figure 37. Equation. Image correlation tracker.

5.5.3 Pedestrian Tracker Integration

In the current system, the tracker is integrated with the PD and classifier to form a closed-loop feedback architecture. The positive outputs from PC, which are the confirmed pedestrian image patches, are fed back to the tracker, and the tracker registers them as new entries and predicts and refines the changing position of ROIs for future frames. The tracked ROIs are removed from the track queue when the following occur:

- ROIs go outside the image boundary.
- Tracking loss occurs.
- Reconfirmation of the pedestrian label from the classifier does not occur during tracking.

Figure 38 shows a pedestrian tracker.

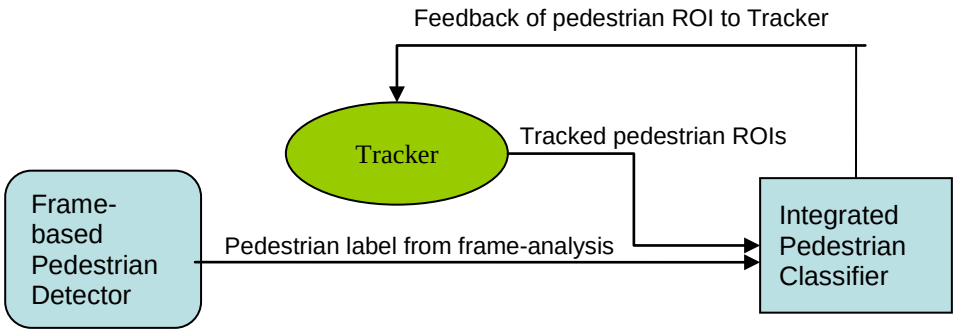


Figure 38. Illustration. Pedestrian tracker data flow.

6. EXPERIMENTS AND RESULTS

The proposed system consists of a stereo rig that is made of off-the-shelf monochrome cameras and a commercial stereo processing board that runs on multicore personal computer environments.[23] The cameras are of standard NTSC automotive grade with 720 × 480 image resolution with a 46-degree field of view. The stereo rig is mounted inside a vehicle (Toyota® Highlander) that also has a dual-quad-core processing unit and electronics to power the computer using the vehicle battery. This test platform allows the researchers to conduct live experiments and collect data for offline processing.

To evaluate system performance, the research team captured and ground truth-marked a number of data sequences in various urban driving scenarios. The testing data included sequences of pedestrians crossing the road, cluttered intersections, and pedestrians darting out from between parked vehicles. The research team also acquired data from publicly available datasets, which are particularly challenging because they have a large number of pedestrians and a crowded urban setting.[30] The research team compared the performance of this system against those of other state-of-the-art systems in the public dataset.[30]

Several image examples of the data collected are provided in figure 39 through figure 51.

Figure 39. Photo. Pedestrians crossing at an intersection during the day under good lighting conditions.

Figure 40. Photo. Pedestrians crossing at an intersection during the day while a vehicle turns right.

Figure 41. Photo. Pedestrian crossing an intersection at night.

Figure 42. Photo. Pedestrians crossing a road at midblock during the evening.

Figure 43. Photo. Pedestrians crossing a road at midblock during the early evening.

Figure 44. Photo. Pedestrians crossing a road at an intersection at night.

Figure 45. Photo. Vehicle driving on the highway.

Figure 46. Photo. Second view of vehicles driving on the highway with tall vertical poles and overhang bridge in the field of view.

Figure 47. Photo. Pedestrians crossing midblock in a multilane urban street with overhang bridge as overlapping background.

Figure 48. Photo. Pedestrian crossing the street and right-turning vehicle in winter.[(12)]

Figure 49. Photo. Pedestrians on the sidewalk in an urban environment during winter.[(12)]

Figure 50. Photo. Pedestrians walking in the roadway near parked vehicles in an urban environment.[12]

Figure 51. Photo. Pedestrians at a crosswalk in front of a vehicle in bright conditions with saturated areas.[12]

6.1 EVALUATION METHODOLOGY

This section briefly discusses the experimental methodology and shows results on selected sequences. The system was evaluated by comparing it to hand-marked ground-truth marked data. For a detailed evaluation, the research team analyzed the performance under the following factors:

- Overall system performance.
- Zone-based (head-on/in-path versus full field of view) performance analysis.
- Performance of each module in the system.

6.2 EXPERIMENTAL RESULTS

The results in this section are presented for typical sequences acquired at the research team's campus, from a Europe dataset, and from a publicly available dataset.[31,12] Overall, the research team captured over 2 h of video data using a vehicle owned by the research team and a vehicle maintained by the research team's automotive tier 1 partner, Autoliv Electronics.

The results are representative of the developed system's performance. It is important to note that for many of these sequences, the FPPF results are somewhat misleading in that the sequences are acquired for the purposes of pedestrian detection and do not have the empty roads that are typical of regular driving scenarios.

For each of the tables shown below, the performance of each key module of the developed system is shown including the stereo-based PD and the detector and classifiers as well as the detector, classifier, and tracker. Results are shown for both in-path pedestrians and all pedestrians in the field of view up to 131.2 ft (40 m).

Additionally, the research team tested the real-time system by driving the vehicle and qualitatively observing true detection and FP performance. The system was tested while driving at speeds of 15 and 30 mi/h (24.15 and 48.3 km/h). Researchers also demonstrated the system multiple times to FHWA personnel at the research team's campus in Princeton, NJ, and at the Turner-Fairbank Highway Research Center in McLean, VA. The following main observations were made during live experiments:

- The frame rate of the developed system under live conditions was between 7.5 and 10 Hz.
- Due to the frame rate performance, the live system performed better at slower speeds (standstill to 15 mi/h (24.15 km/h) than at higher speeds. The offline system showed that if the frame rate were improved to 15 Hz, the performance would improve so that speeds of 30 mi/h (48.3 km/h) could be easily handled.
- At specific operating points of the classifiers in crowded urban environments, the FP performance can be reduced to one or lower every few minutes, but this would reduce the true detection rate. The true and FP numbers for the empirically chosen operating point of

the classifier under different evaluation zones (in path versus full view) are shown in the tables below.

- For highway driving, the FP rate approached zero. This is mainly due to the performance of SC and the vehicle FP rejection classifiers.

Tabulated results are provided in table 3 through table 9. Table 3 results are as follows:

- **Sequence name**: 080613111722_BM-SHJ_cross-in-front (parking lot).
- **Parameters**: In-path (-3.28 to 3.28 ft (-1 to 1 m) from the center of the vehicle).
- **Distance**: 0 to 131.2 ft (0 to 40 m).

Table 3. In-path detection results for sequence 080613111722_BM-SHJ_cross-in-front (parking lot).

Mode	Detection Rate (percent)	FPPF	Number of People
Detector only	100	0.04	70
Detector + classifier	87.14	0	70
Detector + classifier + tracker	95.71	0	70

Table 4 results are as follows:

- **Sequence name**: 080613111722_BM-SHJ_cross-in-front (parking lot).
- **Parameters**: Full field of view.
- **Distance**: 0 to 131.2 ft (0 to 40 m).

Table 4. Full field-of-view detection results for sequence 080613111722_BM-SHJ_ cross-in-front (parking lot).

Mode	Detection Rate (percent)	FPPF	Number of People
Detector only	100	7.09	383
Detector + classifier	87.73	0.36	383
Detector + classifier + tracker	96.87	1.1	383

Table 5 results are as follows:

- **Sequence name**: 080613112933_SHJ_walk_BM_stand_on-side (parking lot).
- **Parameters**: Full field of view.
- **Distance**: 0 to 131.2 ft (0 to 40 m).

Table 5. Full field-of-view detection results for sequence 80613112933_ SHJ_walk_BM_stand_on-side (parking lot).

Mode	Detection Rate (percent)	FPPF	Number of People
Detector-only	95.73	10.06	234
Detector + classifier	90.60	0.54	234
Detector + classifier + tracker	98.29	1.55	234

Table 6 results are as follows:

- **Sequence name**: EuropeTour_Innsbruck.0_20070128_42_SVS_Data.
- **Parameters**: Full field of view.
- **Distance**: 0 to 131.2 ft (0 to 40 m).

Table 6. Full field-of-view detection results for sequence EuropeTour_ Innsbruck.0_20070128_42_SVS_Data.

Mode	Detection Rate (percent)	FPPF	Number of People
Detector-only	90.54	4.436	134
Detector + classifier	72.97	0.58	134
Detector + classifier + tracker	85.14	1.36	134

Table 7 results are as follows:

- **Sequence name**: EuropeTour_Wurzburg.0_20070126_19_SVS_Data.
- **Parameters**: Full field of view.
- **Distance**: 0 to 131.2 ft (0 to 40 m).

Table 7. Full field-of-view detection results for sequence EuropeTour_ Wurzburg.0_20070126_19_SVS_Data.

Mode	Detection Rate (percent)	FPPF	Number of People
Detector-only	86.43	5.35	161
Detector + classifier	70.54	0.98	161
Detector + classifier + tracker	74.03	2.5	161

Table 8 results are as follows:

- **Sequence name**: seq00_rerun (Ess sequence).
- **Parameters**: In-path.
- **Distance**: 0 to 131.2 ft (0 to 40 m).

Table 8. In-path detection results for Sequence seq00_rerun (Ess sequence).

Mode	Detection Rate (percent)	FPPF	Number of People
Detector-only	94.56	0.82	584
Detector + classifier	66.61	0.16	584
Detector + classifier + tracker	92.81	0.45	584

Table 9 results are as follows:

- **Sequence name**: seq00_rerun (Ess sequence).
- **Parameters**: Full field of view.
- **Distance**: 0 to 131.2 ft (0 to 40 m).

Table 9. Full field-of-view detection results for sequence seq00_rerun (Ess sequence).

Mode	Detection Rate (percent)	FPPF	Number of People
Detector-only	91.91	10.78	1,816
Detector + classifier	66.13	1.56	1,816
Detector + classifier + tracker	89.21	3.55	1,816

6.2.1 Comparison Between FHWA Results and Published State-of-the-Art Results

Figure 52 through figure 55 show receiver operating characteristic (ROC) curves illustrating the developed system's performance on four sequences (Seq00, Seq01, Seq02, and Seq03). The figures also show comparisons with another representative approach from literature.[31]

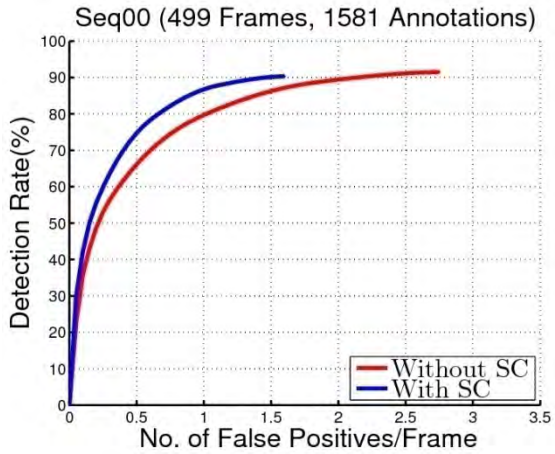


Figure 52. Graph. ROC curves for Seq00.

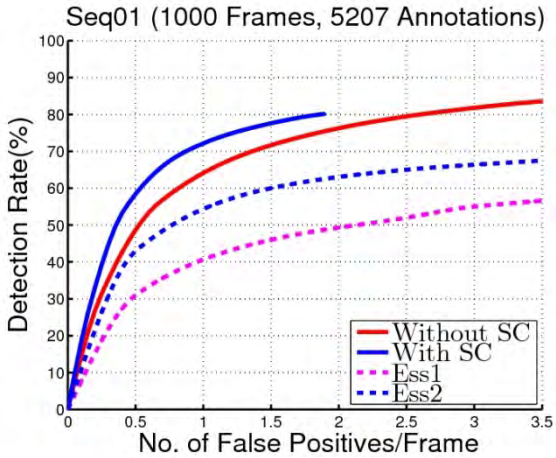


Figure 53. Graph. ROC curves for Seq01.

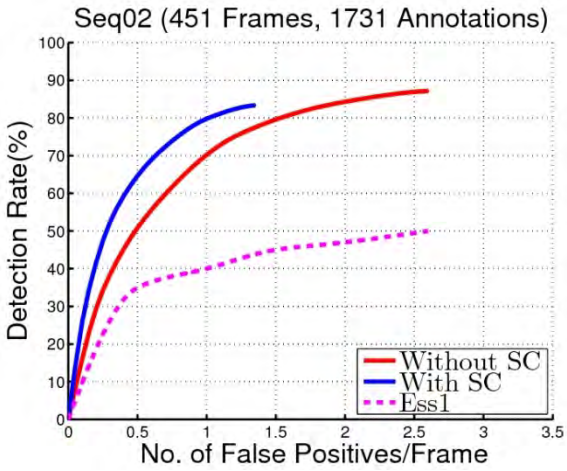


Figure 54. Graph. ROC curves for Seq02.

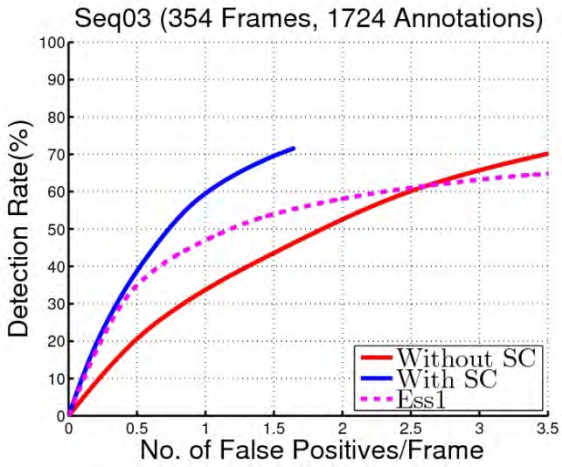


Figure 55. Graph. ROC curves for Seq03.

Example image outputs of the system are provided in figure 56 through figure 65. In the left image in figure 56, magenta and green pixels are detected by the SC as a tall, vertical structure. The image on the right shows the detections that were rejected by the SC in blue. The red rectangles indicate objects that were identified as pedestrians.

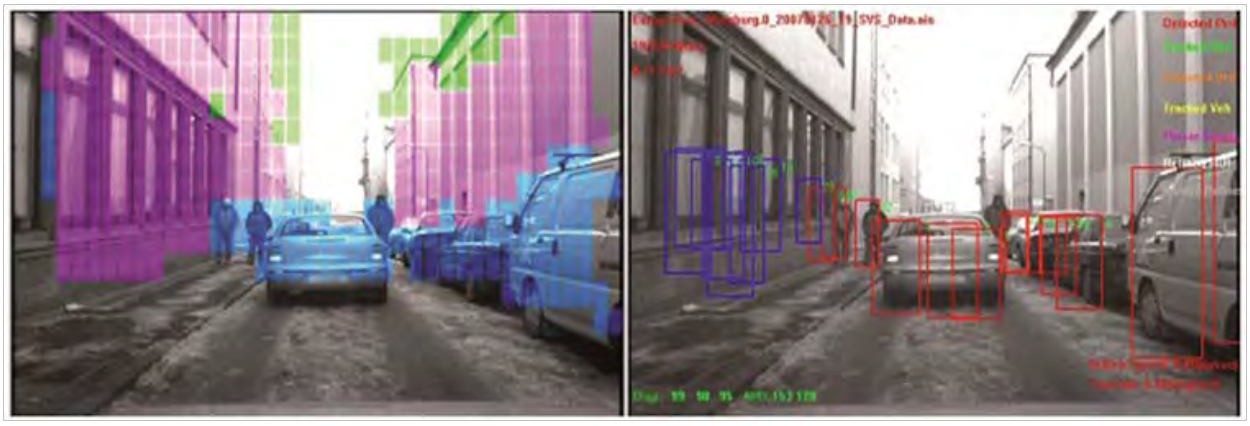


Figure 56. Photo. Sample output from SC in an alleyway.[(12)]

In the image on the left in figure 57, ground pixels are yellow, overhang/tree branch pixels are green, and buildings/tall vertical structure pixels are magenta. Blue pixels indicate regions containing objects that will be further processed by an appearance classifier. In the right image, blue boxes indicate objects rejected by the SC, and white boxes indicate potential pedestrians.

Figure 57. Photo. Sample output from SC in a dense urban scene with pedestrians in the vehicle path.[(12)]

The image on the left in figure 58 shows ground pixels in yellow and tall vertical structure pixels in magenta and green. Pedestrian candidate regions are blue. In the right image, white boxes indicate detected pedestrian candidates, and blue boxes indicate rejected candidates.

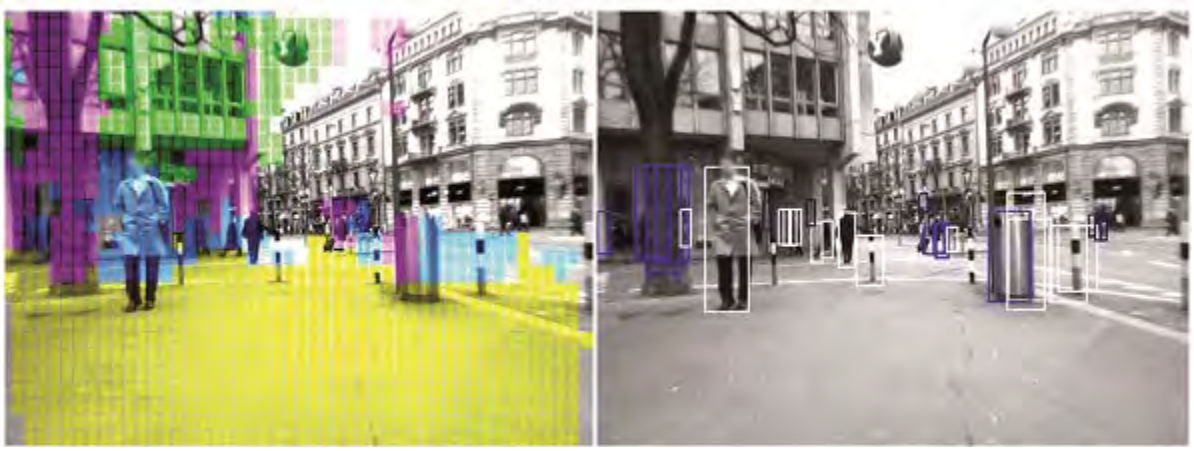


Figure 58. Photo. Sample output from SC in an urban scene with pedestrians at varying distances from the vehicle.[(12)]

In the image on the left in figure 59, the SC correctly rejects the poles and trees in the foreground, which are magenta and green. It also rejects portions of the bicycle parked near the sidewalk while validating the pedestrian detections. In the image on the right, pedestrian detections are shown in white boxes, while rejected candidates have blue boxes around them.

Figure 59. Photo. Sample output from SC in an urban scene with pedestrians entering a building and others in the distance ahead of the vehicle.[(12)]

In figure 60, the SC did not reject the person on the motorcycle or the light post on the median. The image shows tall vertical structures in magenta, overhanging structures in green, and possible pedestrians in blue.

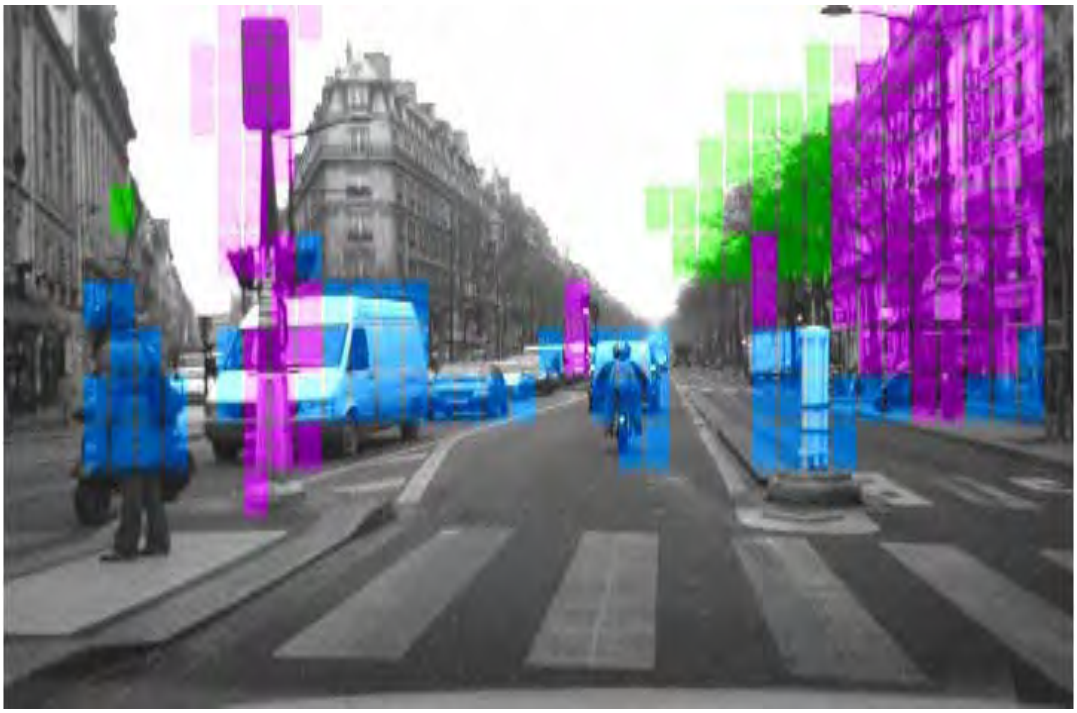


Figure 60. Photo. SC rejecting poles.[12]

Figure 61 through figure 65 show pedestrians detected by the appearance classifier, which are shown by the red boxes.

Figure 61. Photo. Appearance classifier recognizing a pedestrian.[12]

Figure 62. Photo. Appearance classifier output recognizing pedestrians crossing in front of vehicles.

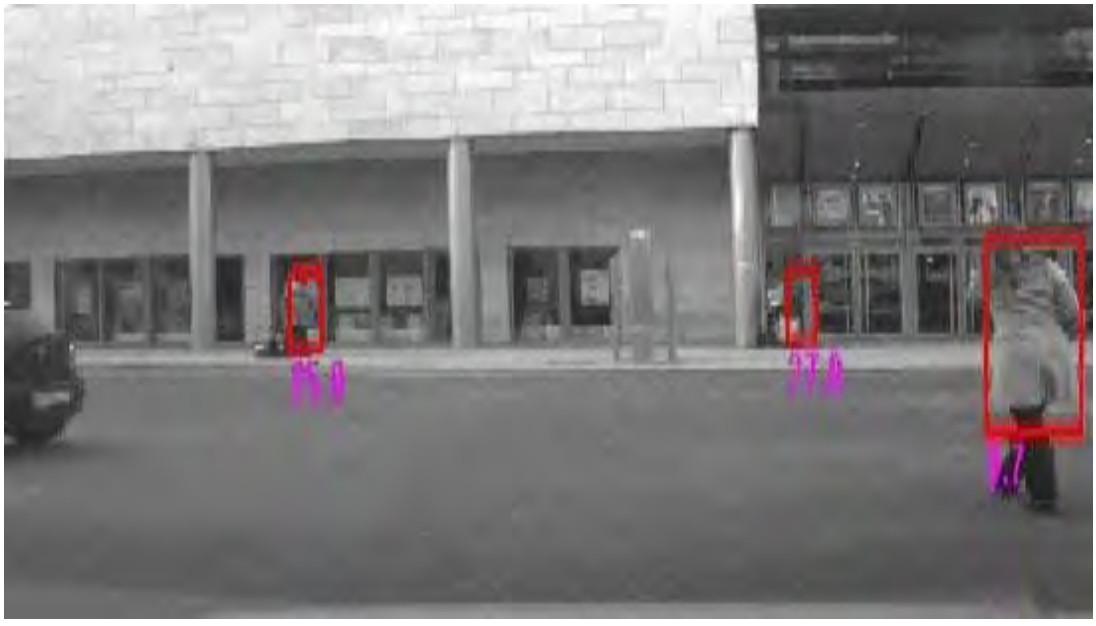
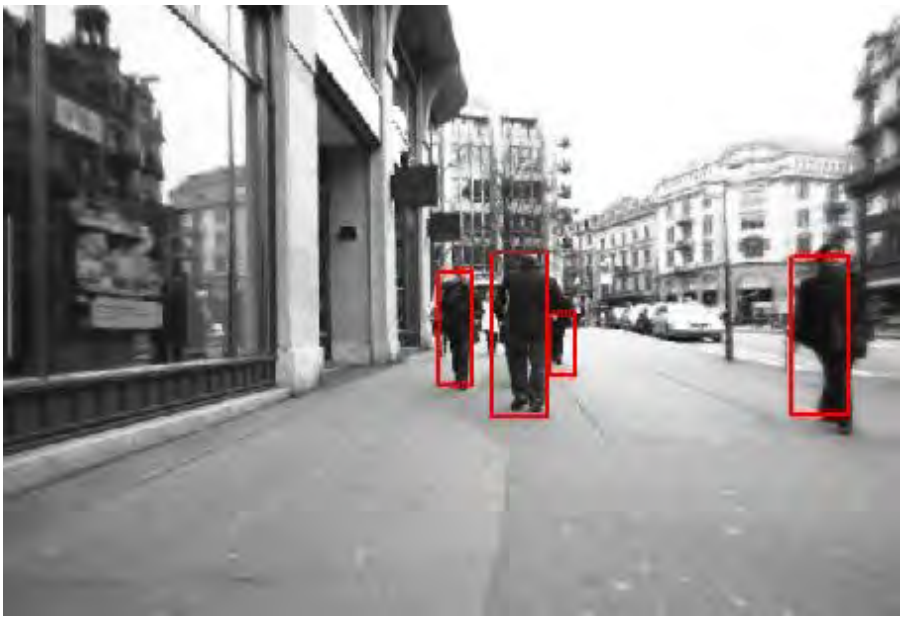


Figure 63. Photo. Appearance classifier output recognizing pedestrians while making a left turn.



©INRIA (See Acknowledgements section)

Figure 64. Photo. Appearance classifier recognizing pedestrians in front of a vehicle in a busy urban street.[12]

©INRIA (See Acknowledgements section)

Figure 65. Photo. Appearance classifier recognizing pedestrians 98.4 ft (30 m) ahead of a vehicle in a busy street.[12]

7. CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

The research team developed a real-time in-vehicle vision-based stereo system that detects and recognizes pedestrians in the camera's field of view. The system uses a layered or hierarchical approach that progressively operates on all or part of the input image data, with each step increasing in computational complexity and reducing the image area that needs to be processed by subsequent steps. The system integrates multiple cues including depth, appearance, and motion. The key steps are as follows:

1. Large-scale object extraction using stereo depth templates.

2. SC recognition of multiple classes including ground, buildings, trees, and poles and separation of those classes from vehicles and pedestrians.

3. Appearance classification using a cascade of classifiers that explicitly recognizes pedestrians and discriminates against other objects such as vehicles and bushes.

4. Pedestrian tracking using shape and appearance matching.

Based on offline and live experiments using a Toyota® Highlander with a stereo camera head and a personal computer processing unit, the following conclusions were made:

- The system achieves state-of-the-art performance for detection rate and FP rate when compared to other published results.

- The FP rate achieved by the system is not low enough for deployment as a stand-alone system. This performance argues for either the use of an additional sensor (e.g., using radar/light detection and ranging or reducing the horizontal field of view of the stereo camera to achieve production-level performance).

- The system needs further optimization to improve its performance to 15 Hz or higher. A higher frame rate is needed so that the system can be used on vehicles traveling at speeds higher than 30 mi/h (48.3 km/h).

If successful, the following recommendations for future work could lead to a commercially viable system:

- Enforce optimizations to increase throughput and reduce system latency.

Implement the developed system on an embedded platform. Potential candidates include the Acadia II™ application-specific integrated circuit in combination with a field-programmable gate array (FPGA) or the automotive-grade multiple digital signal processor and FPGA system jointly developed by Autoliv Electronics and Sarnoff Corporation.

- Improve the classification performance to further reduce FPs. Most of the FPs are from specific objects. While some progress has been made to remove these types of consistent FPs, there is a need for additional development to categorize these detections using supervised or unsupervised learning techniques and to build more focused classifier cascades that can reject them.

- Test and create enhancements to enable the system to operate in off-road conditions and construction zones.

APPENDIX

This appendix describes the user interface that was developed for the real-time layered object recognition system for pedestrian collision sensing software system.

The graphic user interface (GUI) consists of a main page in which subsequent modules can be customized and run. For each of these modules (i.e., PD, classification, etc.), tab pages are defined, which can be subsequently customized if users click the respective tab. A screenshot of the main page is shown in figure 66. It is the first interface that users see when operating the system.

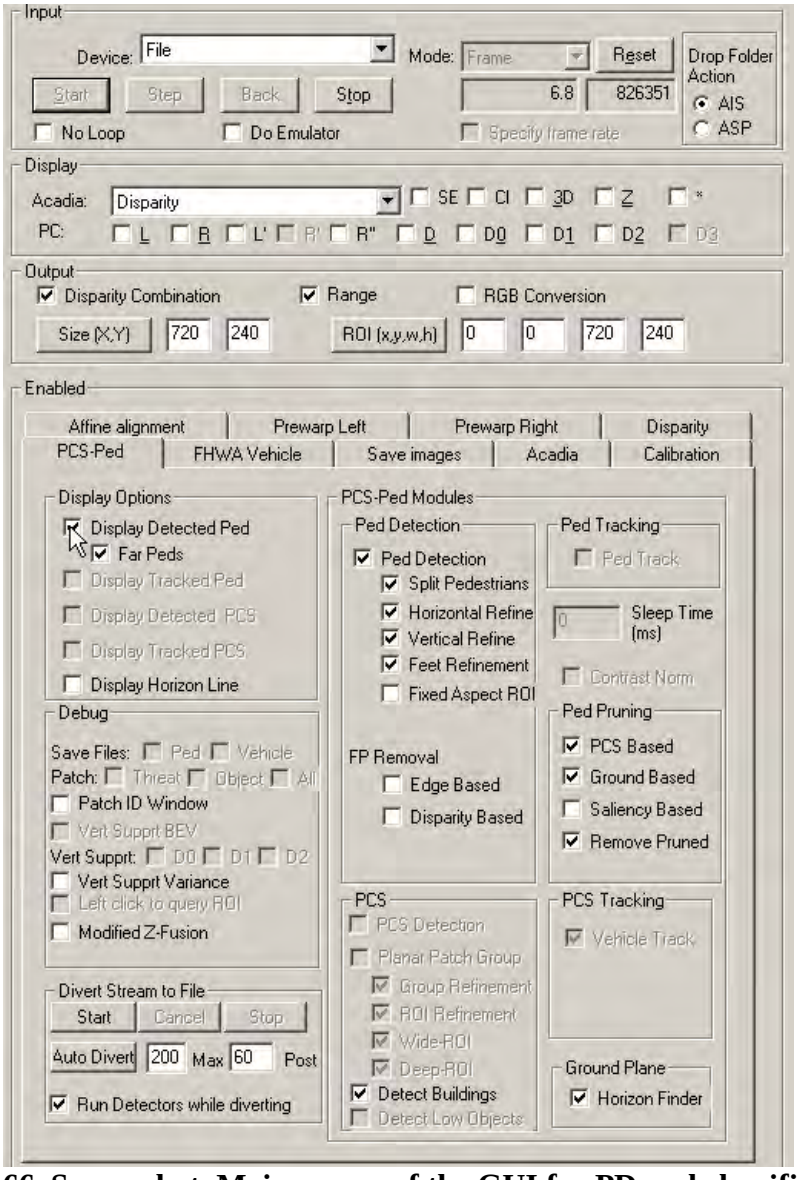


Figure 66. Screenshot. Main screen of the GUI for PD and classification.

This appendix illustrates each of the options through captured screenshots. The two key properties exposed by GUI are controls for the stereo-based detection system and controls for PC. The controls for the stereo-based detection system help find ROIs in an input image. The controls for PC help prune the detections and reduce the number of false pedestrians returned by the system. The arrow in figure 67 indicates the selection option to display all of the pedestrian candidates detected by the system prior to classification.

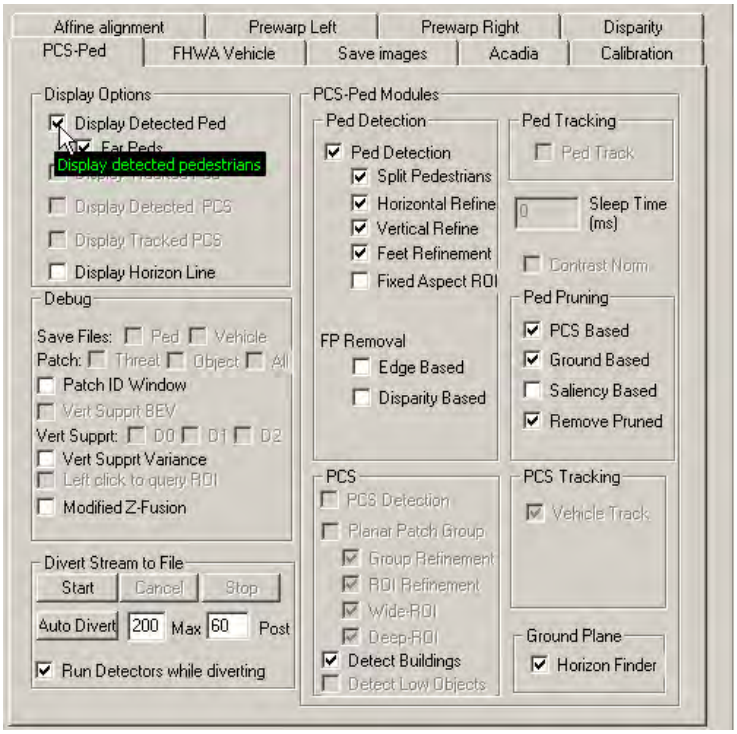


Figure 67. Screenshot. PD interface—display all detected pedestrian candidates.

Figure 68 shows the PCS-Ped tab of the GUI interface. The arrow indicates the selection option to display all pedestrian candidates 82 ft (25 m) away from the vehicle detected by the system prior to classification.

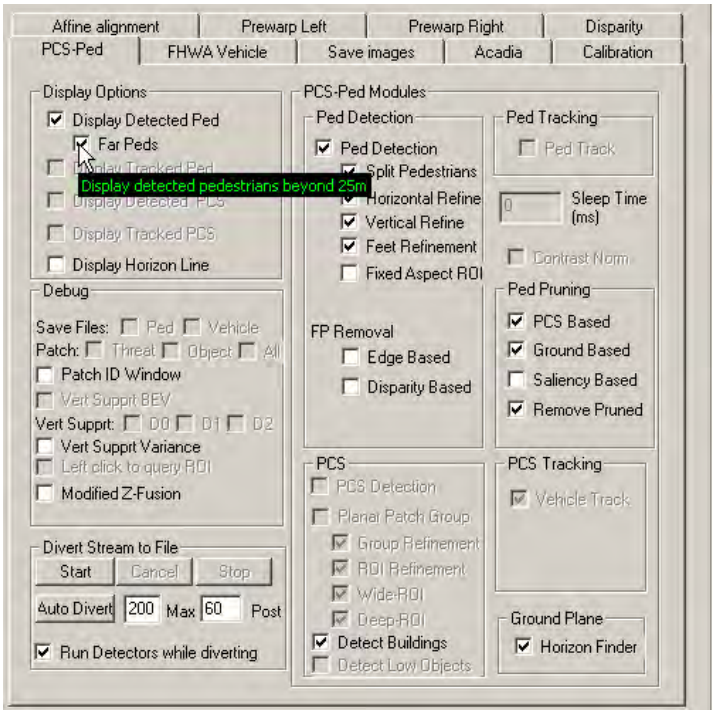


Figure 68. Screenshot. PD interface—PCS-Ped tab with option selected to display detected pedestrians.

Figure 69 also shows the PCS-Ped tab within the GUI interface. The arrow indicates the selection option to display the horizon line estimated by the system. This option is a byproduct of the ground plane estimator.

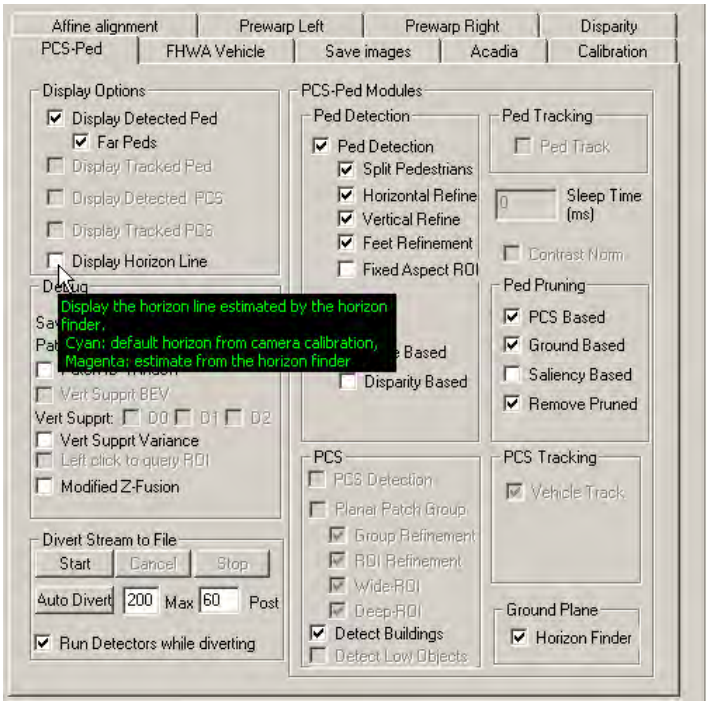


Figure 69. Screenshot. PD interface—PCS-Ped tab with option selected to display horizon line estimated by the system.

Figure 70 shows the PCS-Ped tab of the GUI interface. The arrow indicates the selection option to display the SC output.

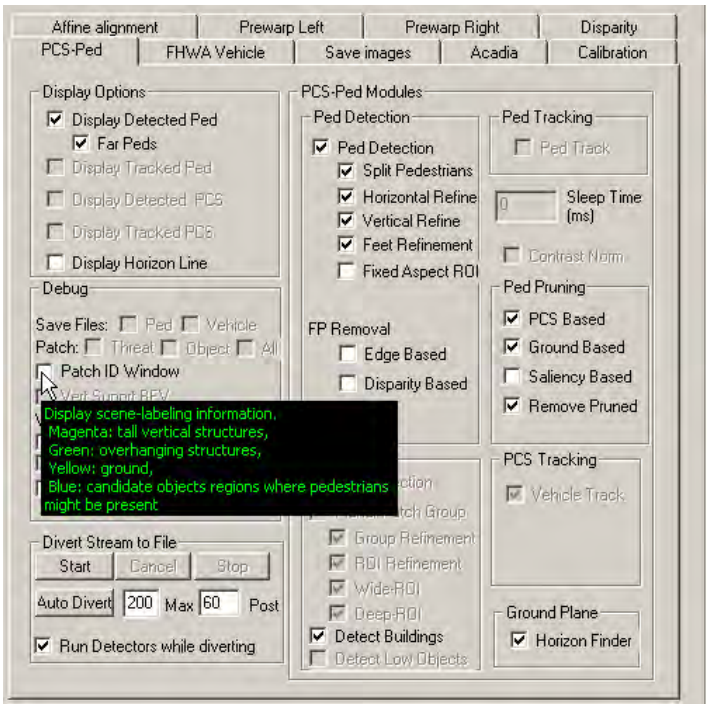


Figure 70. Screenshot. PD interface—PCS-Ped tab with option selected to display the SC output.

Figure 71 shows the PCS-Ped tab of the GUI interface. The arrow indicates the selection option to display an intermediate VSH output of SC.

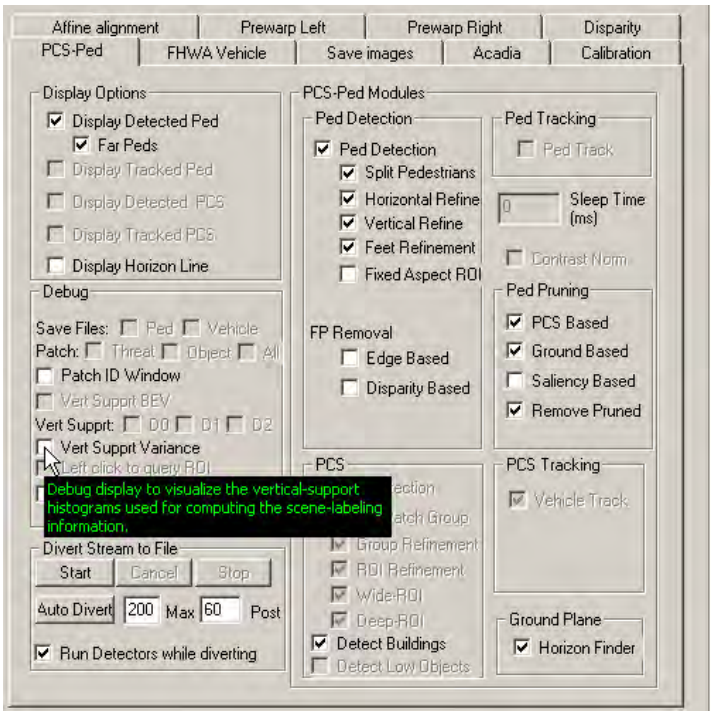


Figure 71. Screenshot. PD interface—PCS-Ped tab with option selected to display an intermediate VSH output of SC.

Figure 72 shows the PCS-Ped tab of the GUI interface. The arrow indicates the selection option to display the depth/disparity map generated by the stereo algorithm.

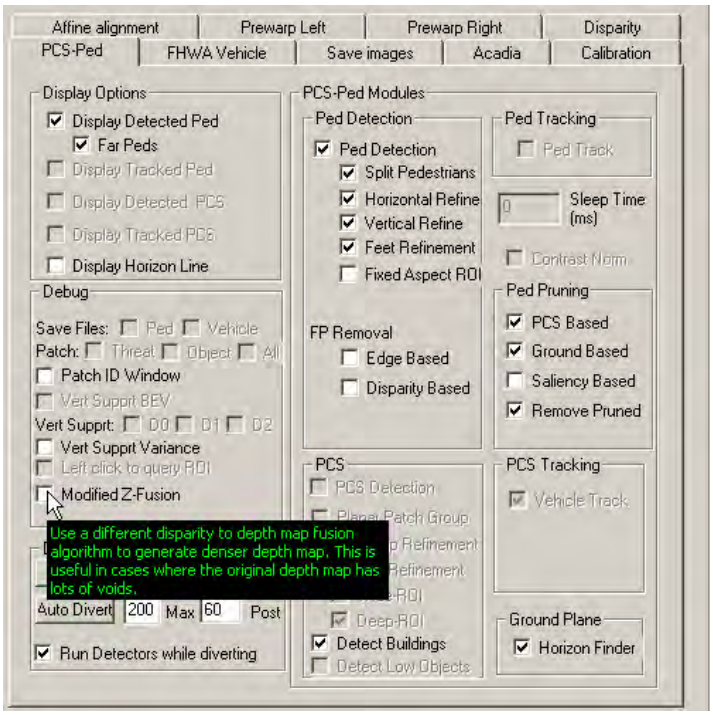


Figure 72. Screenshot. PD interface—PCS-Ped tab with option selected to display depth/disparity map.

Figure 73 shows the PCS-Ped tab of the GUI interface. The arrow indicates the selection option to capture stereo data for temporary storage in the personal computer.

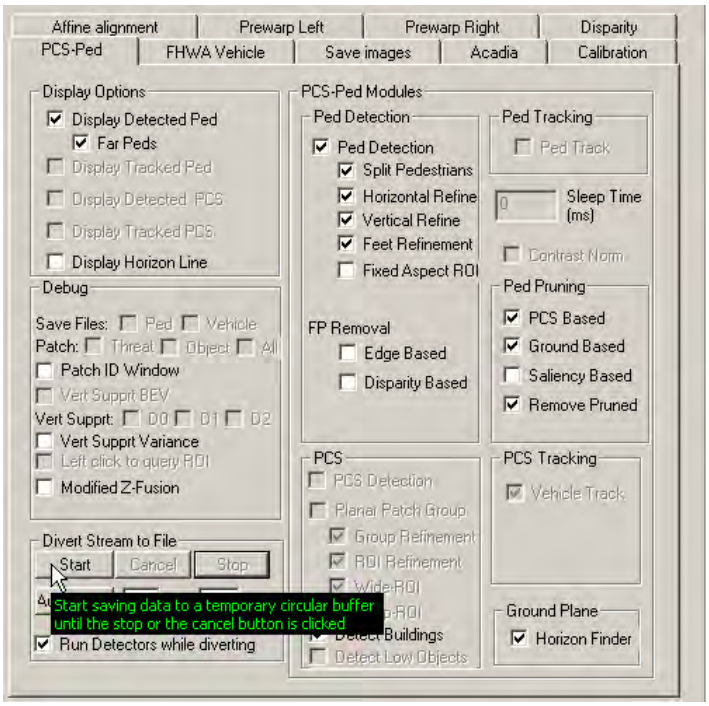


Figure 73. Screenshot. PD interface—PCS-Ped tab with option selected to capture stereo data for temporary storage.

Figure 74 shows the PCS-Ped tab of the GUI interface. The arrow indicates the selection option to cancel saving of stereo data and clear the temporary store in the personal computer.

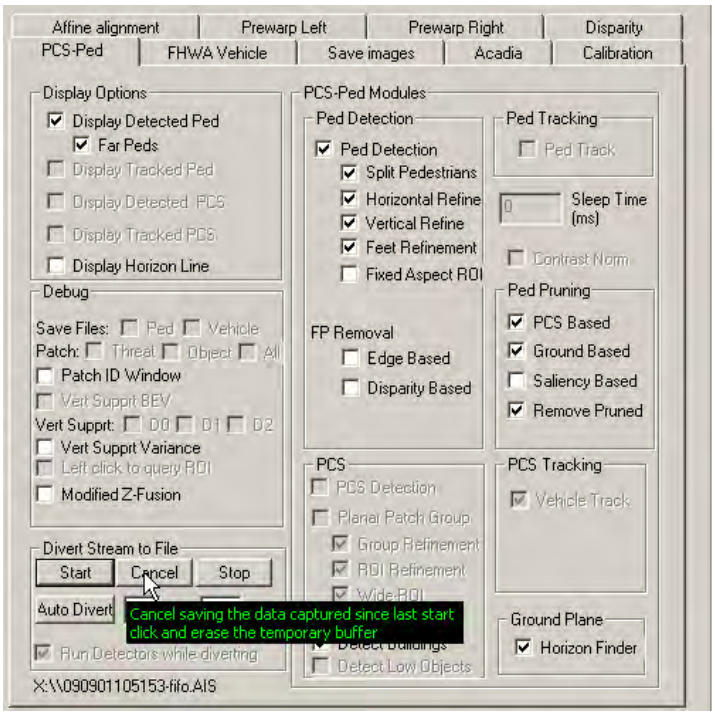


Figure 74. Screenshot. PD interface—PCS-Ped tab with option selected to cancel saving of stereo data and clear temporary store.

Figure 75 shows the PCS-Ped tab of the GUI interface. The arrow indicates the selection option to stop capture and store the captured stereo data to permanent storage on the disk.

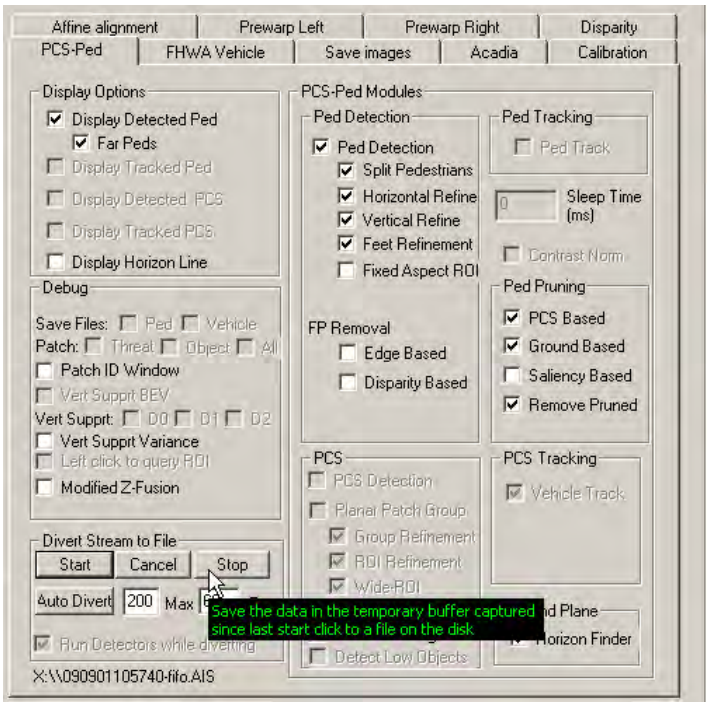


Figure 75. Screenshot. PD interface—PCS-Ped tab with option selected to stop capture and store captured stereo data to permanent storage.

Figure 76 shows the PCS-Ped tab of the GUI interface. The arrow indicates the selection option to automatically divert data to a file on disk whenever a pedestrian is detected.

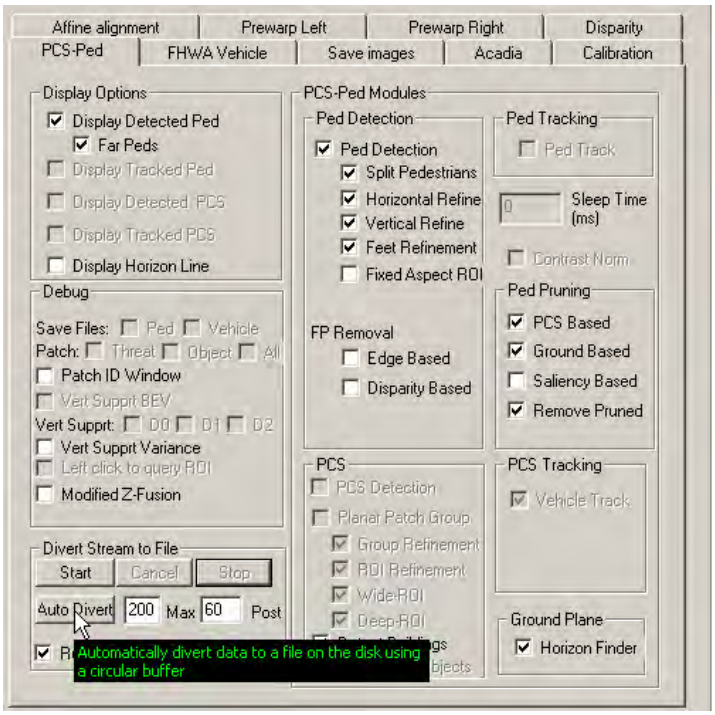


Figure 76. Screenshot. PD interface—PCS-Ped tab with option to automatically divert data to a file whenever a pedestrian is detected.

Figure 77 shows the PCS-Ped tab of the GUI interface. The arrow indicates the selection option to define the maximum number of frames that are maintained in temporary storage during the automatic divert of data to disk.

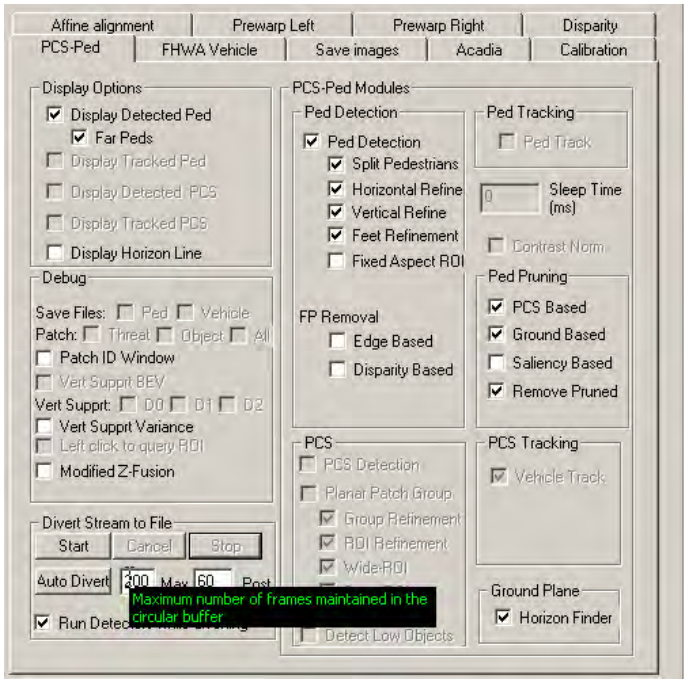


Figure 77. Screenshot. PD interface—PCS-Ped tab with option selected to define maximum number of frames maintained in temporary storage during automatic divert of data.

Figure 78 shows the PCS-Ped tab of the GUI interface. The arrow indicates the selection option that specifies the number of additional video frames saved to disk after “Stop” is selected during data storage to the disk.

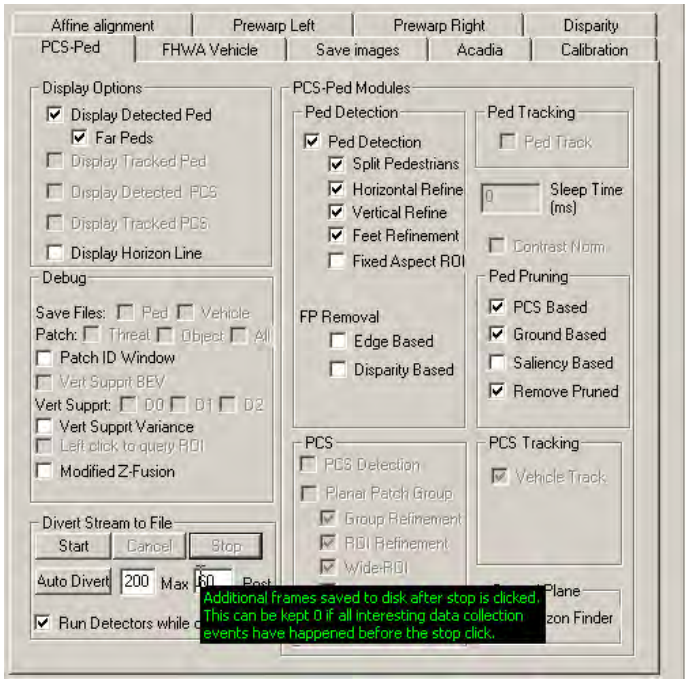


Figure 78. Screenshot. PD interface—PCS-Ped tab with option selected that specifies number of additional video frames saved to disk.

Figure 79 shows the PCS-Ped tab of the GUI interface. The arrow indicates the selection option that specifies whether the PD algorithms should operate while data are being stored to the disk.

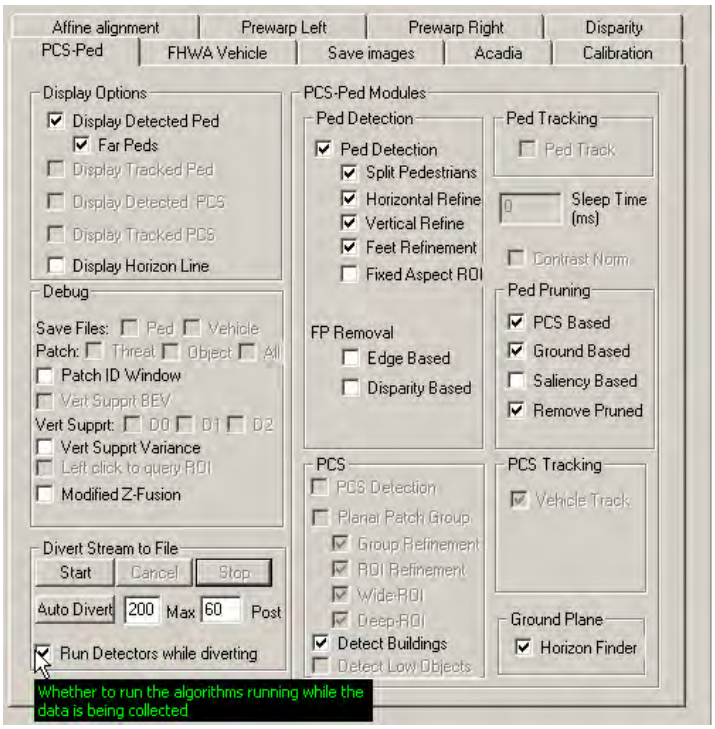


Figure 79. Screenshot. PD interface—PCS-Ped tab with option selected that specifies whether PD algorithms should operate while data being stored.

Figure 80 shows the PCS-Ped tab of the GUI interface. The arrow indicates the selection option that enables the PD algorithm to run in the live system.

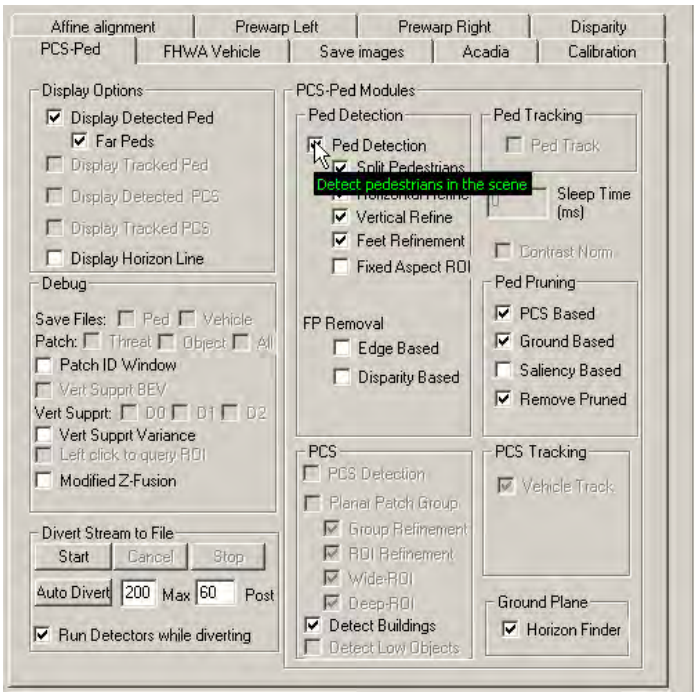


Figure 80. Screenshot. PD interface—PCS-Ped tab with option selected that enables PD algorithm to run in live system.

Figure 81 shows the PCS-Ped tab of the GUI interface. The arrow indicates the selection option that enables the PD algorithm to split wide object detections into multiple pedestrian candidates. This is turned on by default and used to resolve detections within groups of pedestrians observed together by the cameras.

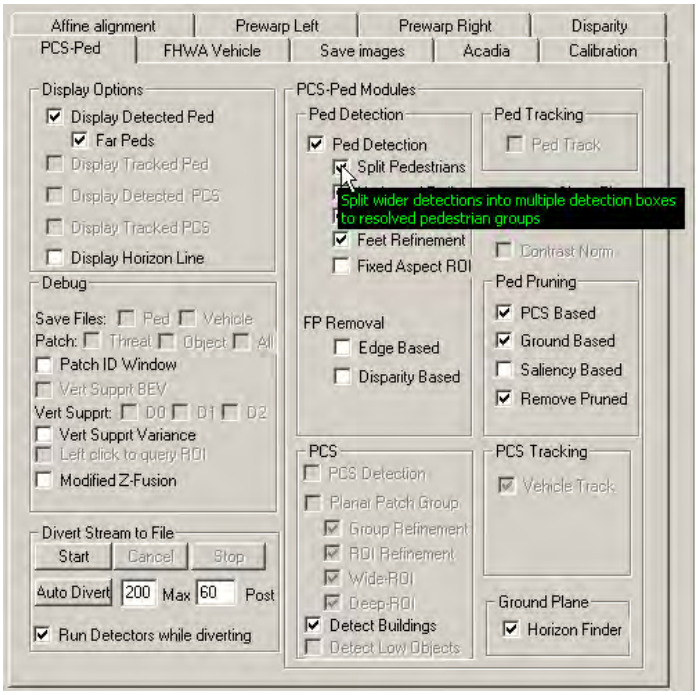


Figure 81. Screenshot. PD interface—PCS-Ped tab with option selected that enables PD algorithm to split wide object detections into multiple pedestrian candidates.

Figure 82 shows the PCS-Ped tab of the GUI interface. The arrow indicates the selection option that enables the PD algorithm to refine the horizontal placement of the initial detection box using depth and edge data.

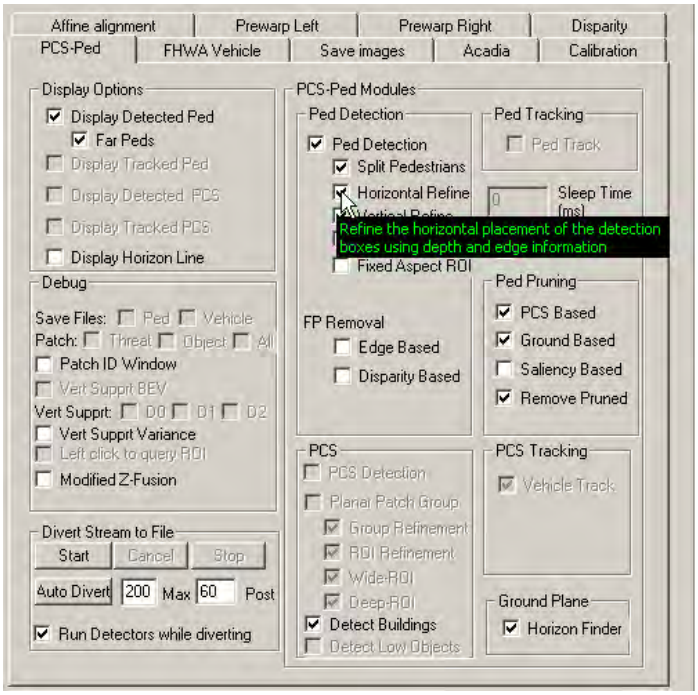


Figure 82. Screenshot. PD interface—PCS-Ped tab with option selected that enables PD algorithm to refine horizontal placement of initial detection box.

Figure 83 shows the PCS-Ped tab of the GUI interface. The arrow indicates the selection option that enables the PD algorithm to refine the vertical placement of the initial detection box using depth and edge data.

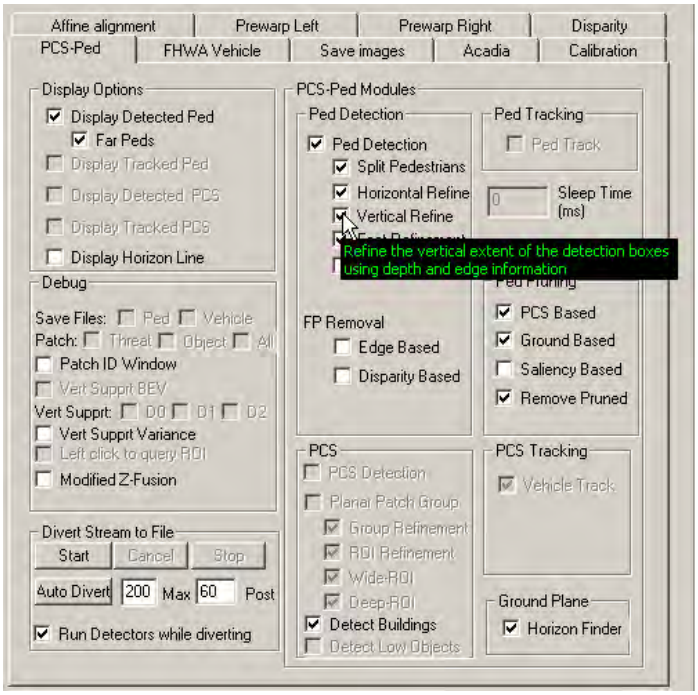


Figure 83. Screenshot. PD interface—PCS-Ped tab with option selected that enables PD algorithm to refine vertical placement of initial detection box.

Figure 84 shows the PCS-Ped tab of the GUI interface. The arrow indicates the selection option that enables the PD algorithm to use the ground plane estimate to better locate the foot location of a detected pedestrian candidate.

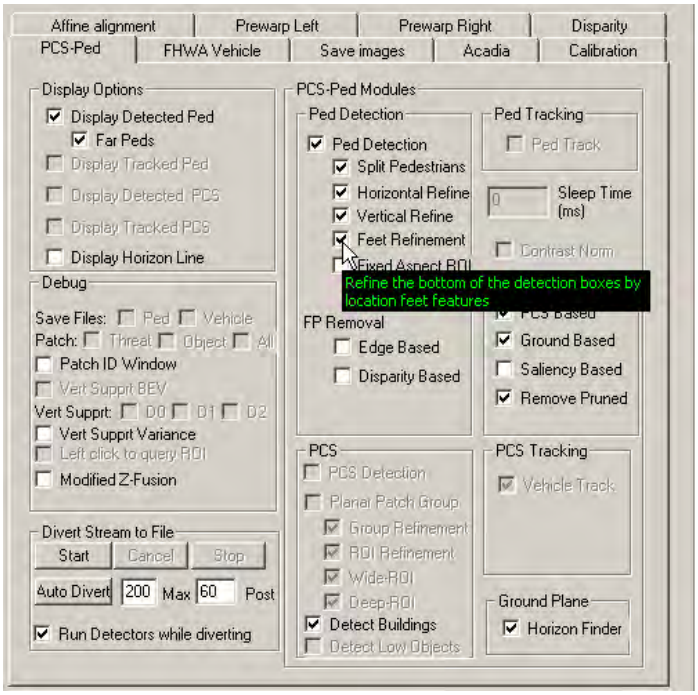


Figure 84. Screenshot. PD interface—PCS-Ped tab with option selected that enables PD algorithm to use ground plane estimate to better locate pedestrians.

Figure 85 shows the PCS-Ped tab of the GUI interface. The arrow indicates the selection option that enables the PD algorithm to maintain a fixed aspect ratio when detection boxes are refined.

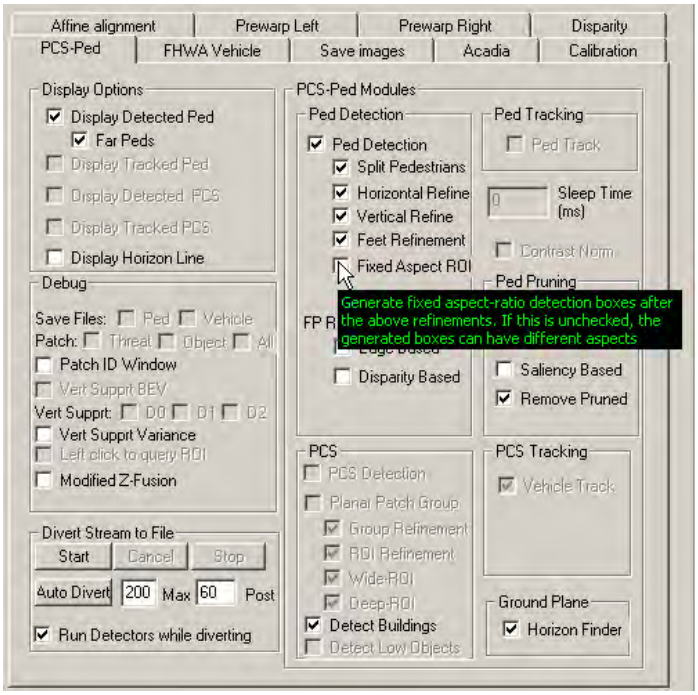


Figure 85. Screenshot. PD interface—PCS-Ped tab with option selected that enables PD algorithm to maintain a fixed aspect ratio when detection boxes are refined.

Figure 86 shows the PCS-Ped tab of the GUI interface. The arrow indicates the selection option that enables the PD algorithm to use image edge information to reject FPs.

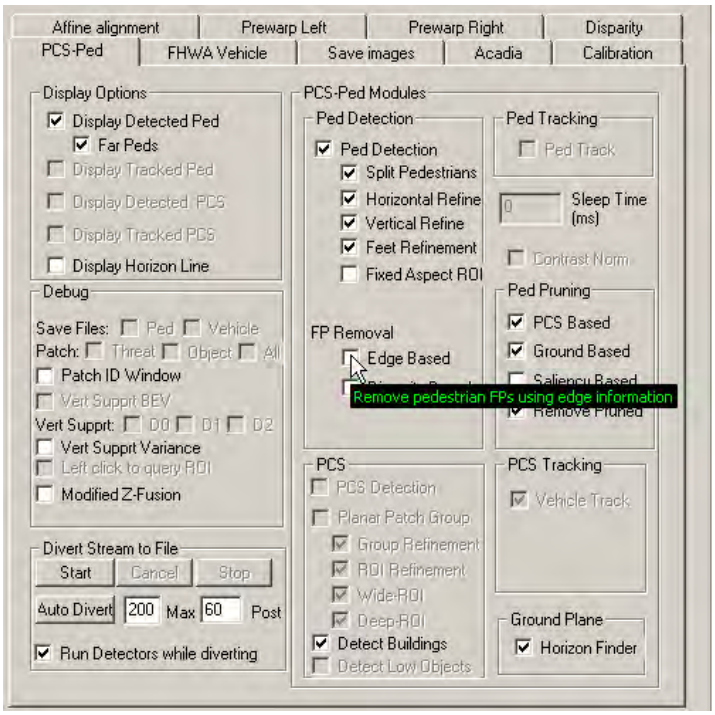


Figure 86. Screenshot. PD interface—PCS-Ped tab with option selected that enables PD algorithm to use image edge information to reject FPs.

Figure 87 shows the PCS-Ped tab of the GUI interface. The arrow indicates the selection option that enables the PD algorithm to use image depth information to reject FPs.

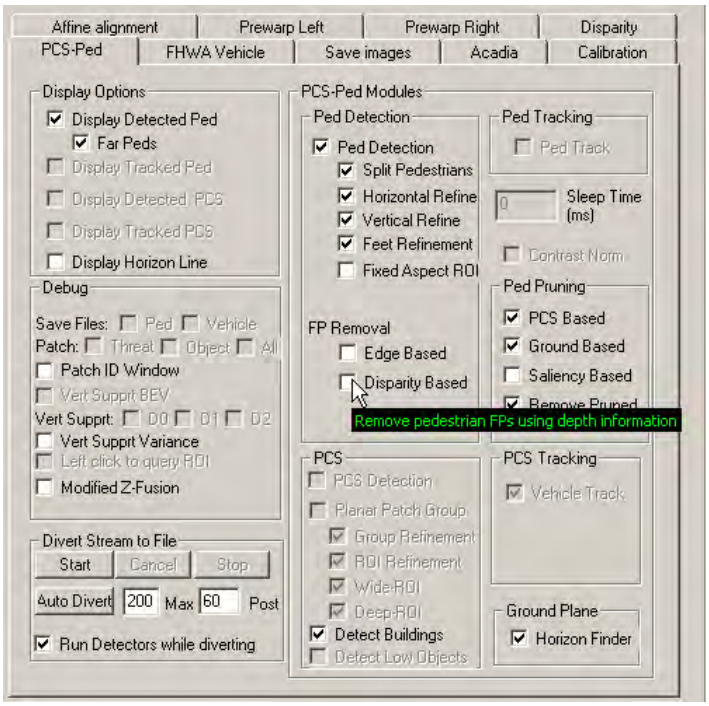


Figure 87. Screenshot. PD interface—PCS-Ped tab with option selected that enables PD algorithm to use image depth information to reject FPs.

Figure 88 shows the PCS-Ped tab of the GUI interface. The arrow indicates the selection option that enables the algorithm to use the SC algorithm to detect tall vertical structures (i.e., buildings, trees, and poles).

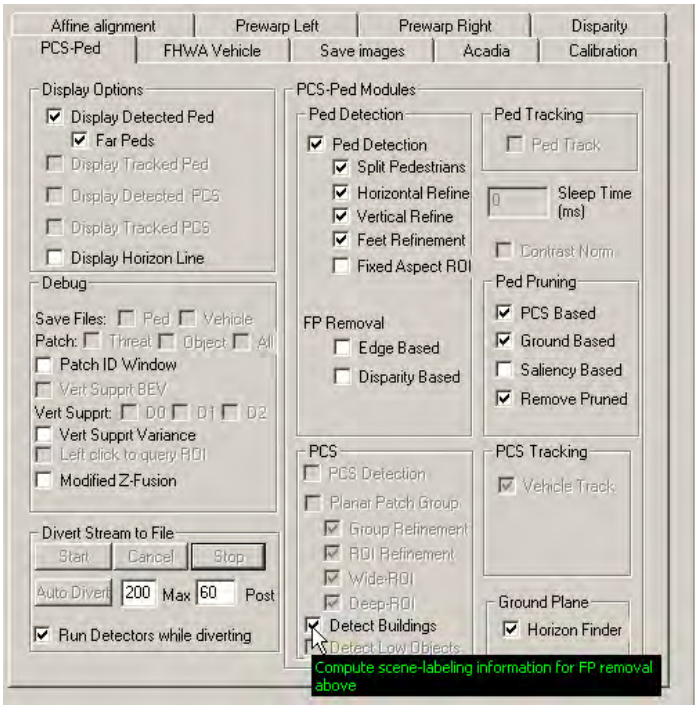


Figure 88. Screenshot. PD interface—PCS-Ped tab with option selected that enables PD algorithm to use SC algorithm to detect tall vertical structures.

Figure 89 shows the PCS-Ped tab of the GUI interface. The arrow indicates the selection option that enables the PD algorithm to reject FPs as indicated by the SC algorithm.

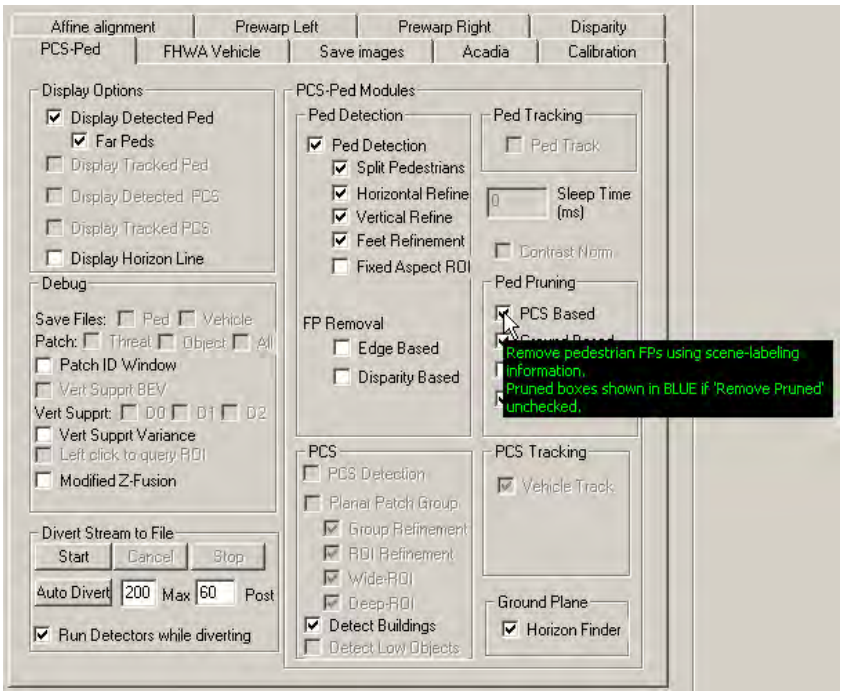


Figure 89. Screenshot. PD interface—PCS-Ped tab with option selected that enables PD algorithm to reject FPs as indicated by SC algorithm.

Figure 90 shows the PCS-Ped tab of the GUI interface. The arrow indicates the selection option that enables the PD algorithm to use ground plane and horizon information to reject FPs.

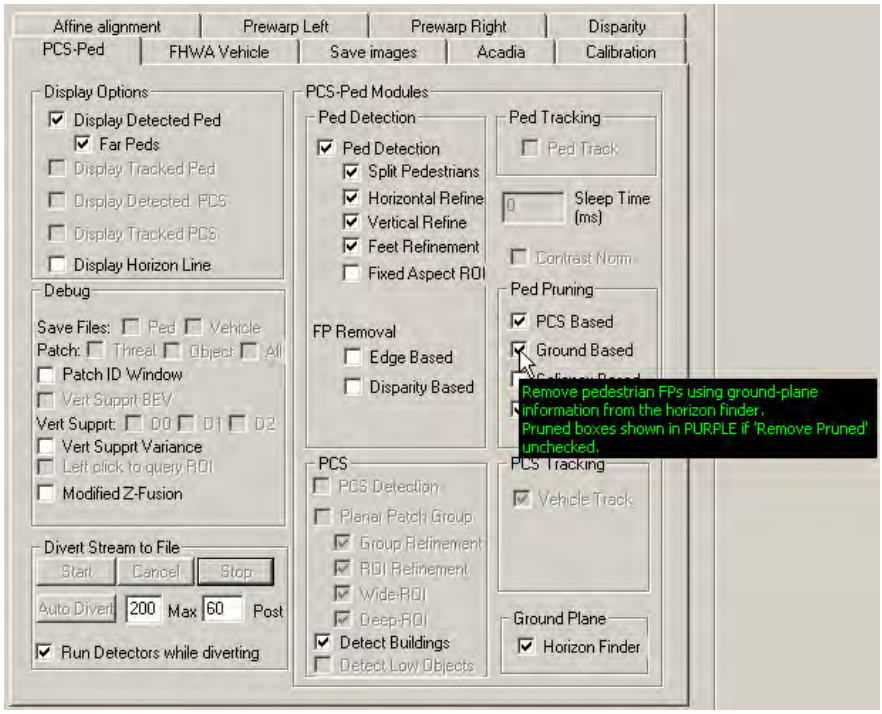


Figure 90. Screenshot. PD interface—PCS-Ped tab with option selected that enables PD algorithm to use ground plane and horizon information to reject FPs.

Figure 91 shows the PCS-Ped tab of the GUI interface. The arrow indicates the selection option that enables the PD algorithm to use image saliency information to reject FPs.

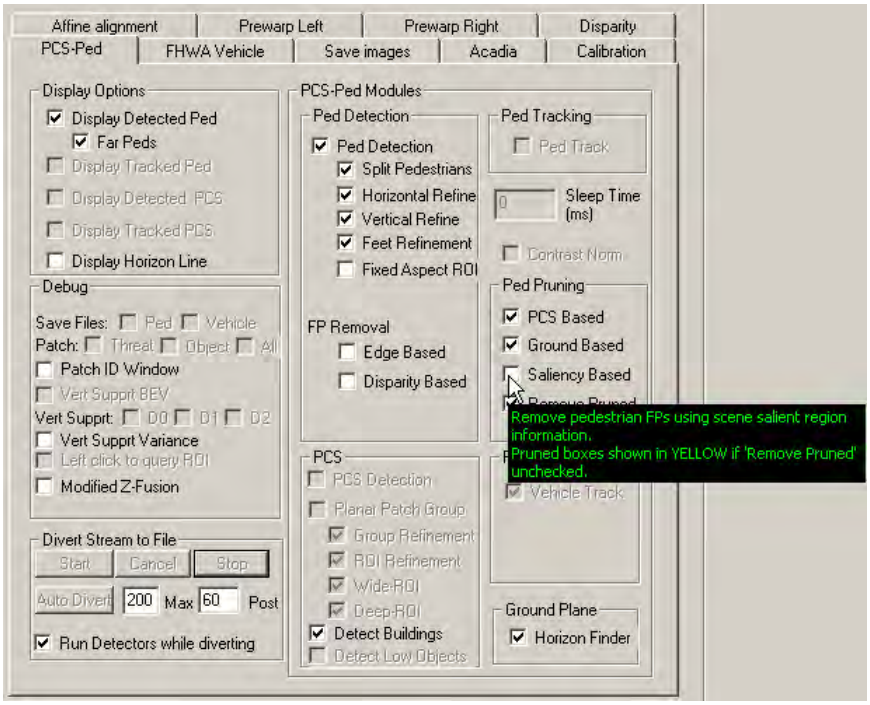


Figure 91. Screenshot. PD interface—PCS-Ped tab with option selected that enables PD algorithm to use image saliency information to reject FPs.

Figure 92 shows the PCS-Ped tab of the GUI interface. The arrow indicates the selection option that enables the PD algorithm to reject FPs detected by the three previous rejection algorithms.

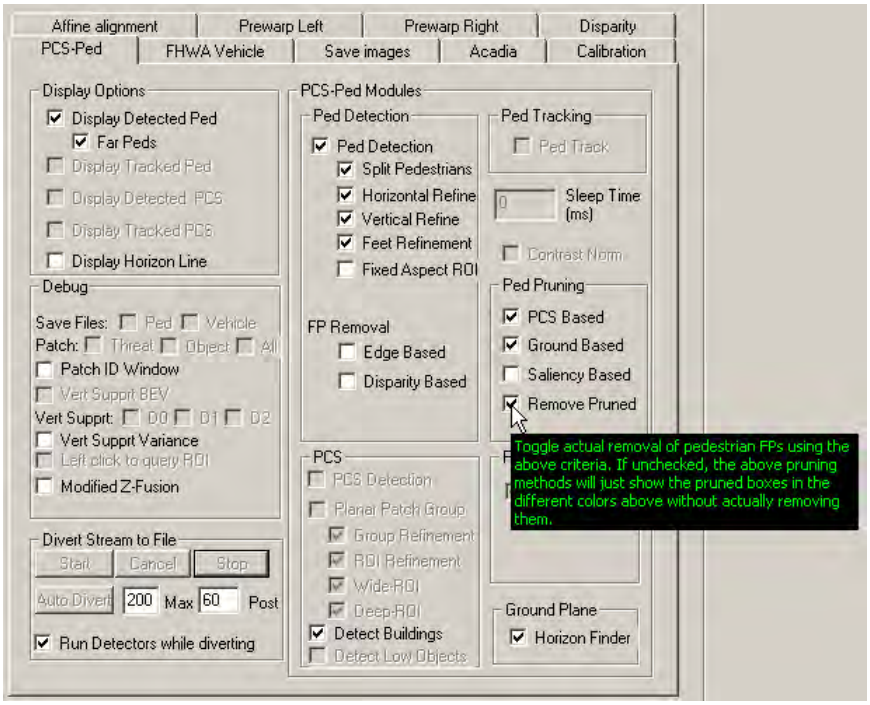


Figure 92. Screenshot. PD interface—PCS-Ped tab with option selected that enables PD algorithm to reject FPs detected by three previous rejection algorithms.

Figure 93 shows the PCS-Ped tab of the GUI interface. The arrow indicates the selection option that enables the PD algorithm to compute the ground plane in the scene. It is used for ground-based FP rejection.

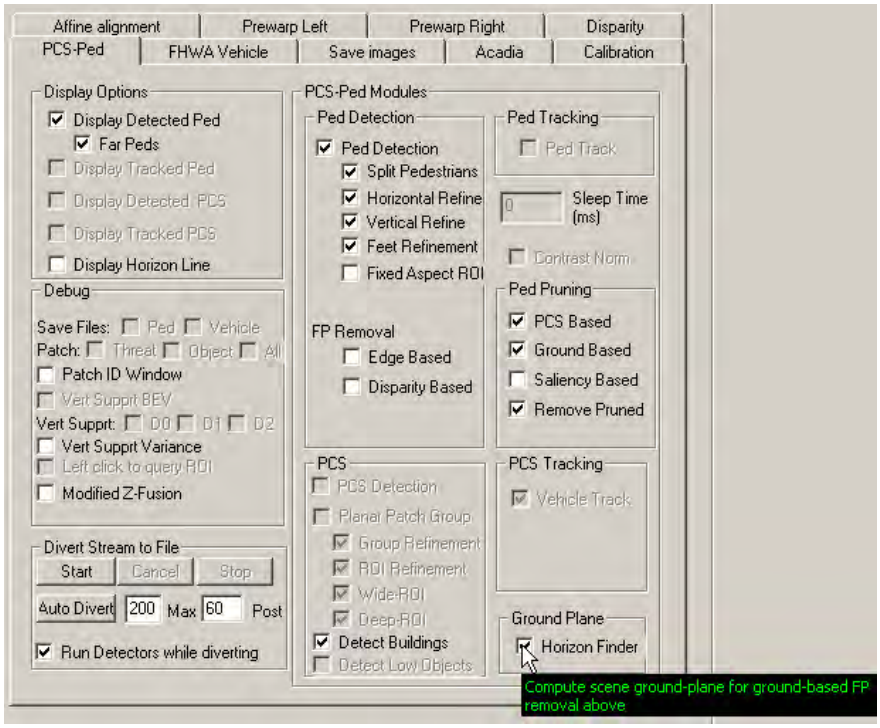


Figure 93. Screenshot. PD interface—PCS-Ped tab with option selected that enables PD algorithm to compute the ground plane in the scene.

Figure 94 through figure 105 show the PC interface. The circled boxes indicate the options that specify the classifier search range around a detection box. It is computed by multiplying the box width and height by the ratios.

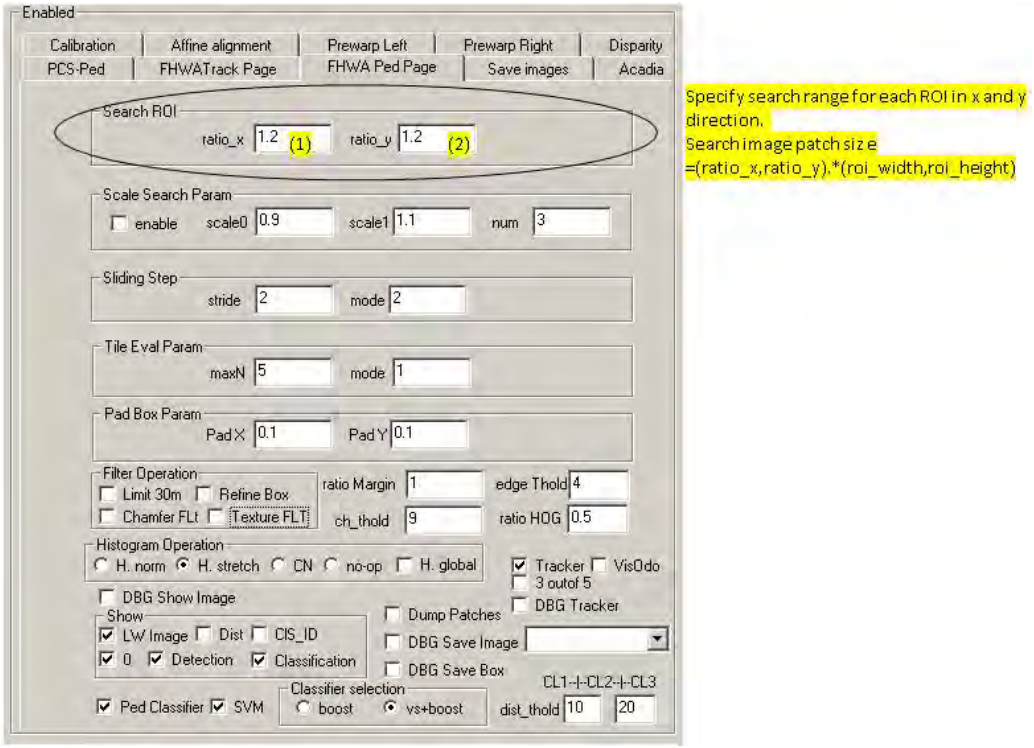


Figure 94. Screenshot. PC interface specifying search range for each ROI in the X and Y directions.

The circled boxes in figure 95 indicate the following selection options: (1) enable scale search, (2) minimum scale, (3) maximum scale, and (4) number of scales to search.

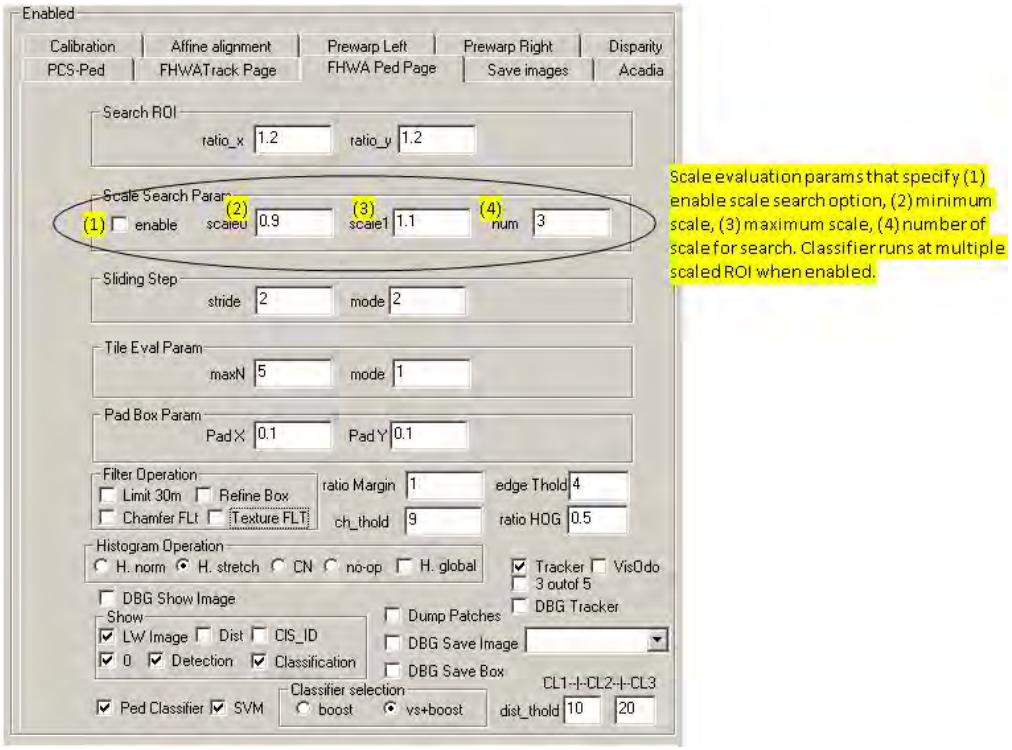


Figure 95. Screenshot. PC interface showing scale evaluation parameters.

The circled boxes in figure 96 indicate the following selection options: (1) classifier search step size in pixels and (2) classifier mode. The classifier mode option is not used in the developed system.

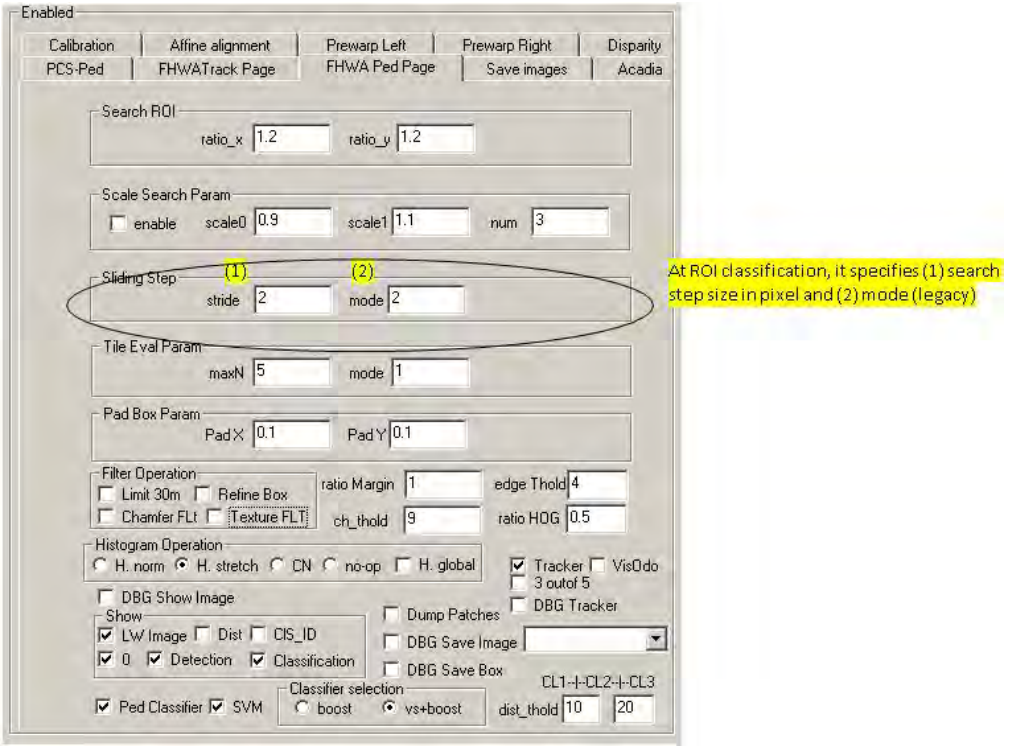


Figure 96. Screenshot. PC interface showing specifications at ROI classification.

The circled boxes in figure 97 indicate legacy classifier parameters that are used for debugging.

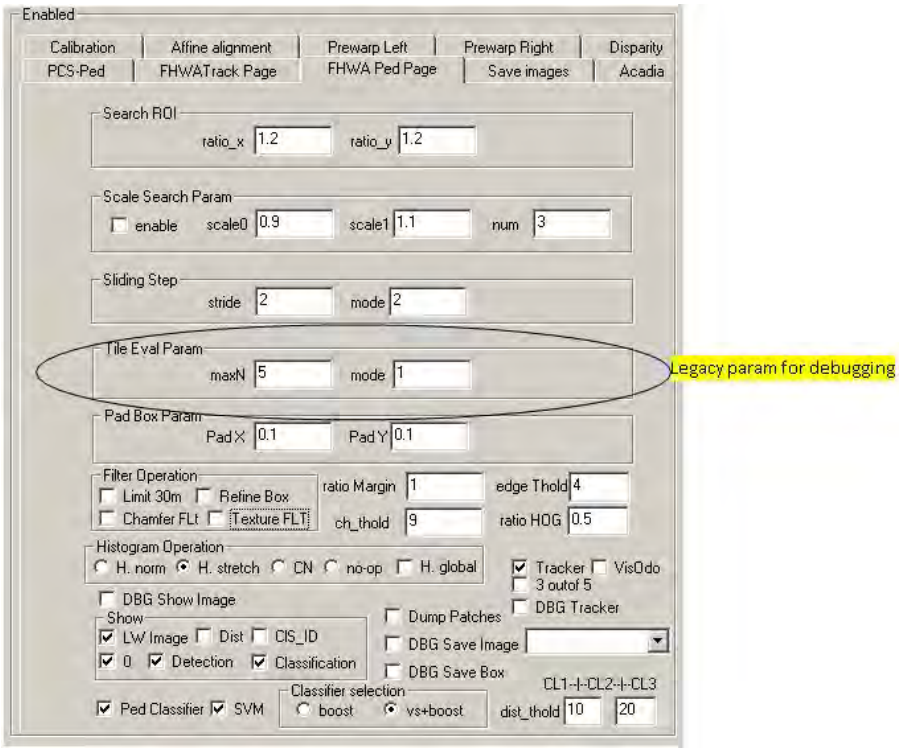


Figure 97. Screenshot. PC interface indicating legacy parameters for debugging.

The circled boxes in figure 98 indicate the selection options where the operator can specify the size of the window padding around a detection box. This is used to create a larger classifier ROI.

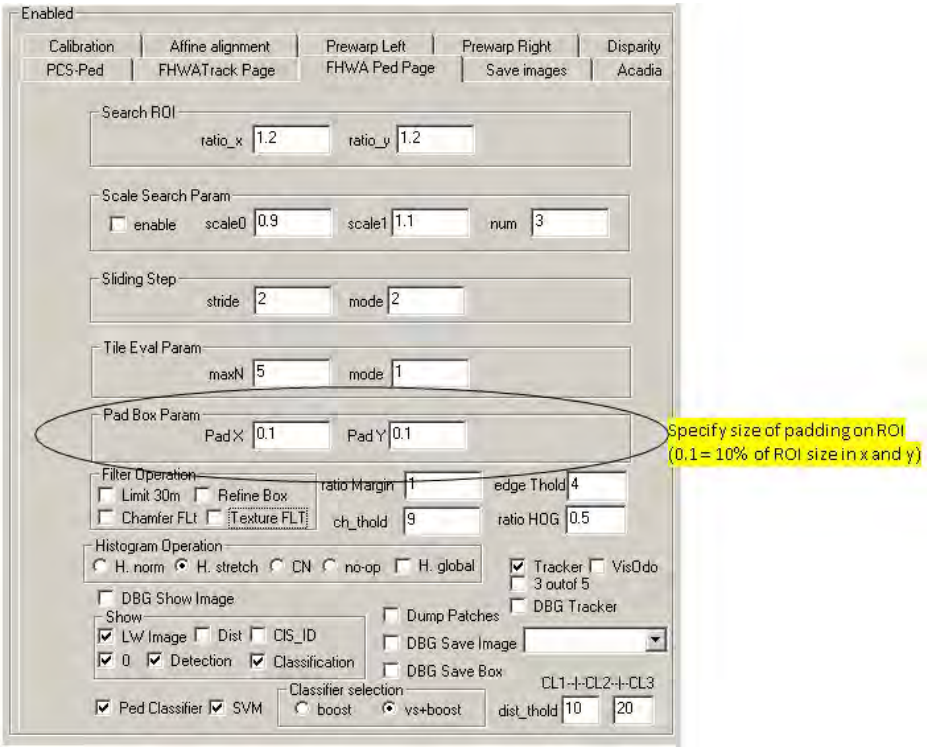


Figure 98. Screenshot. PC interface specifying size of padding around a detection box.

The circled options in figure 99 indicate the following filter operations: (1) limit pedestrian detection to 98.4 ft (30 m), (2) use contour-based ROI refinement, (3) use chamfer score-based filter, and (4) use texture filter. Note that (3) and (4) use additional filters for FP rejections.

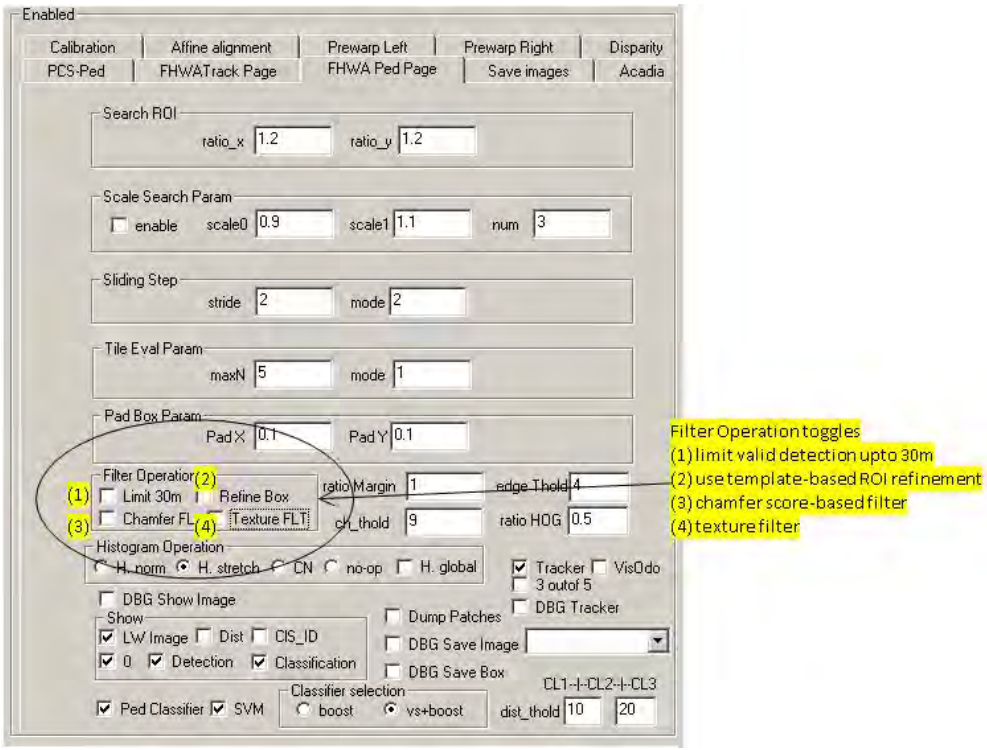


Figure 99. Screenshot. PC interface showing filter options.

The circled options in figure 100 indicate the selection options for image enhancement prior to classification. The four options include (1) histogram equalization, (2) histogram stretch, (3) contrast normalization, and (4) no operation.

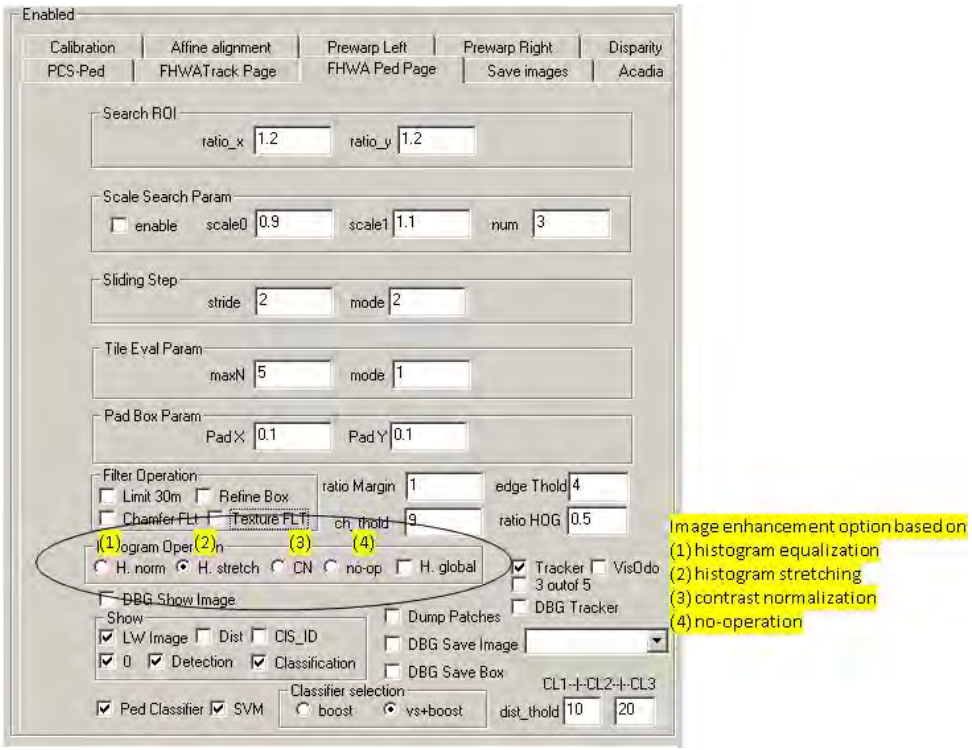


Figure 100. Screenshot. PC interface showing selection options for image enhancement prior to classification.

The circled options in figure 101 indicate selection options for classifier output display. The options are as follows: (1) show display window, (2) show distance, (3) show ID, (4) show overlay detection with classification boxes, (5) show detection boxes, and (6) show classified pedestrians.

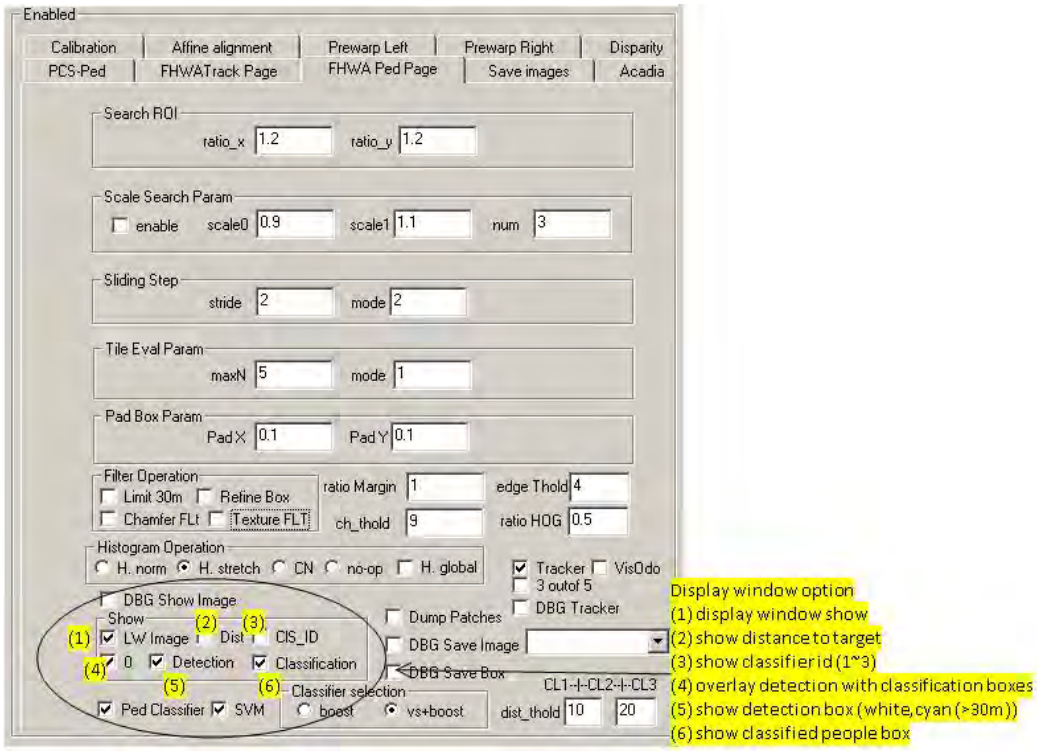


Figure 101. Screenshot. PC interface showing options for classifier output display.

The circled options in figure 102 indicate the selection options to run the PC and a post-processing SVM classifier for bush rejection.

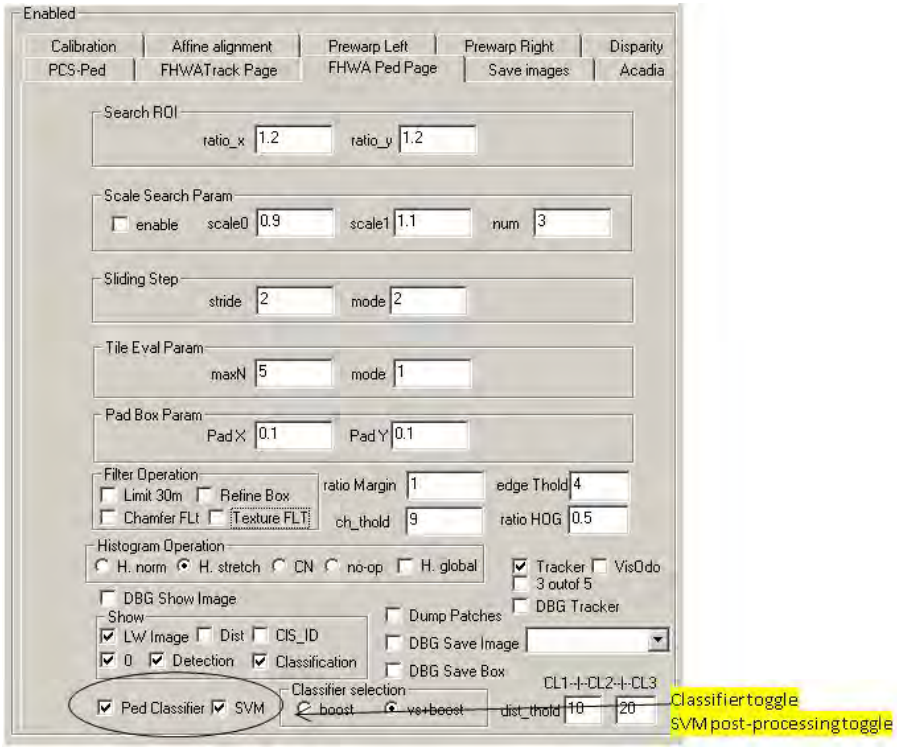


Figure 102. Screenshot. PC interface showing selection options to run PC and a post-processing SVM classifier for bush rejection.

The circled options in figure 103 indicate options to select a HOG AdaBoost classifier (Boost) or a contour plus HOG AdaBoost classifier (VS + Boost).

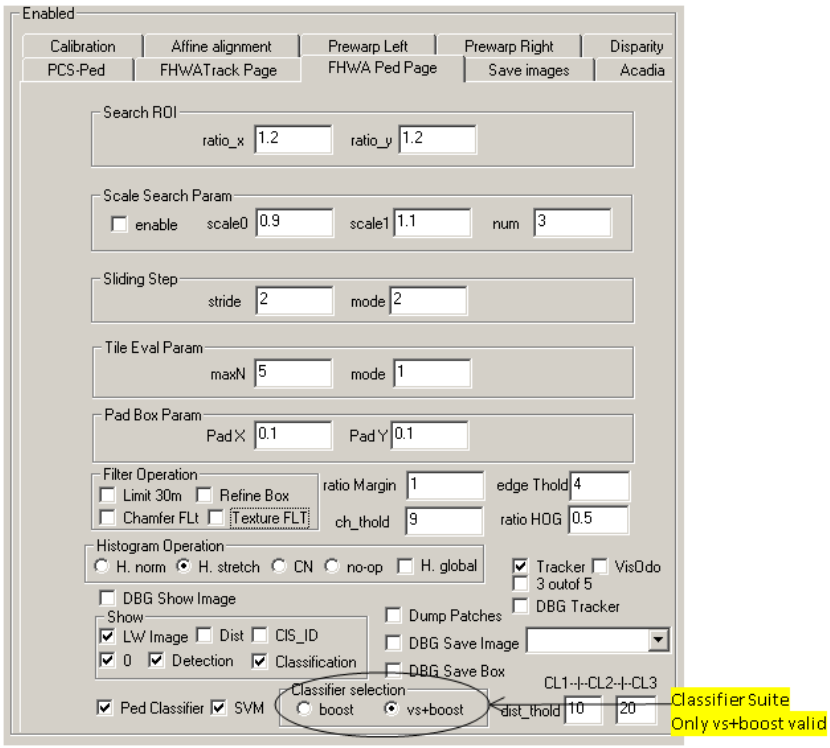


Figure 103. Screenshot. PC interface showing options to select a HOG AdaBoost classifier or contour plus HOG AdaBoost classifier.

The boxes circled in figure 104 indicate the selection options used to decide the distance ranges at which the contour + HOG classifier (0 to 32.8 ft (0 to 10 m)) and the basic HOG classifiers (32.8 to 65.6 ft (10 to 20 m) and over 65.6 ft (20 m)) are used.

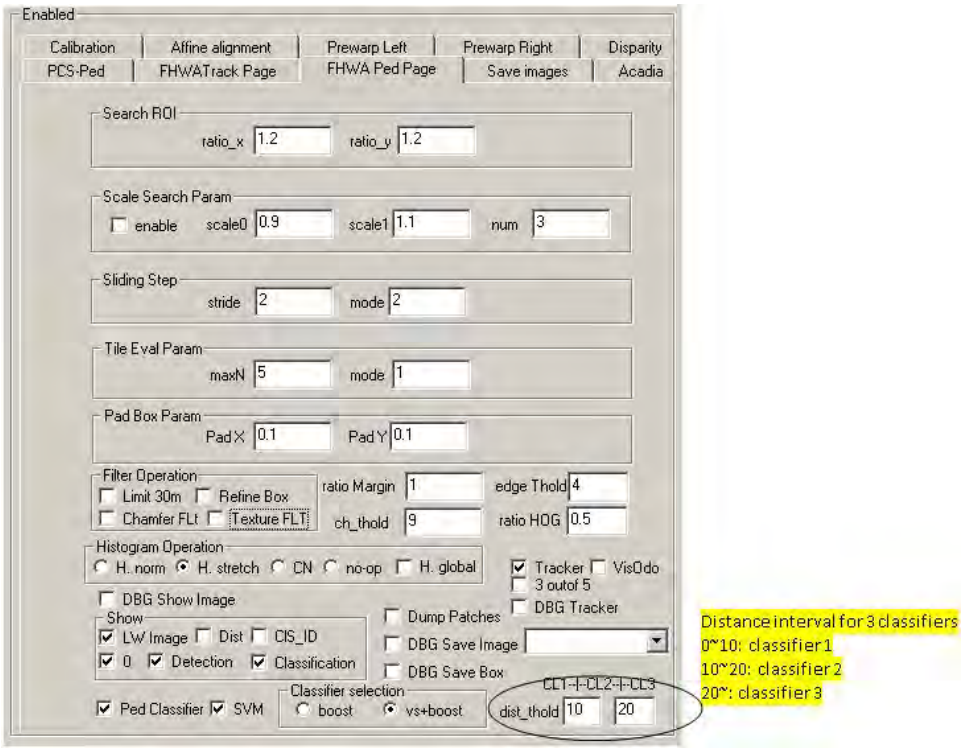


Figure 104. Screenshot. PC interface showing selection options to decide distance ranges for three classifiers.

The circled options in figure 105 indicate the following classifier debugging options:
(1) save detected image patches to a file, (2) save the current image with detection boxes, and (3) save detection statistics to a file.

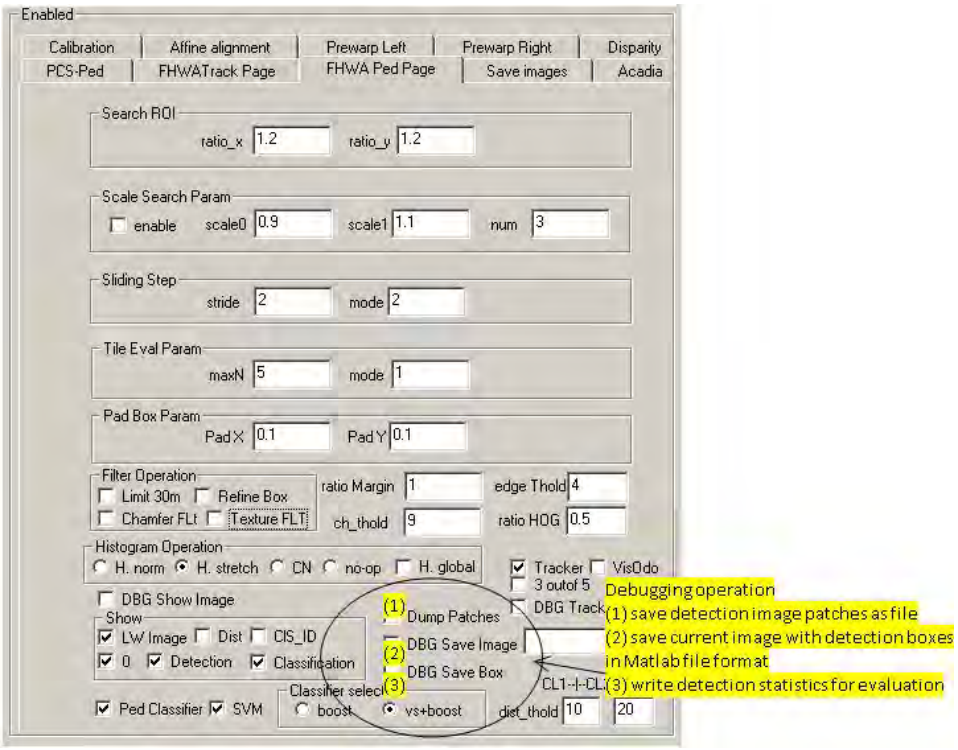


Figure 105. Screenshot. PC interface showing classifier debugging options.

The options circled in figure 106 indicate the following tracker interface options: (1) enable the tracker, (2) apply a heuristic that looks for consistent defects in three out of five frames, (3) show debugging labels for the tracker, and (4) enable the egomotion estimator.

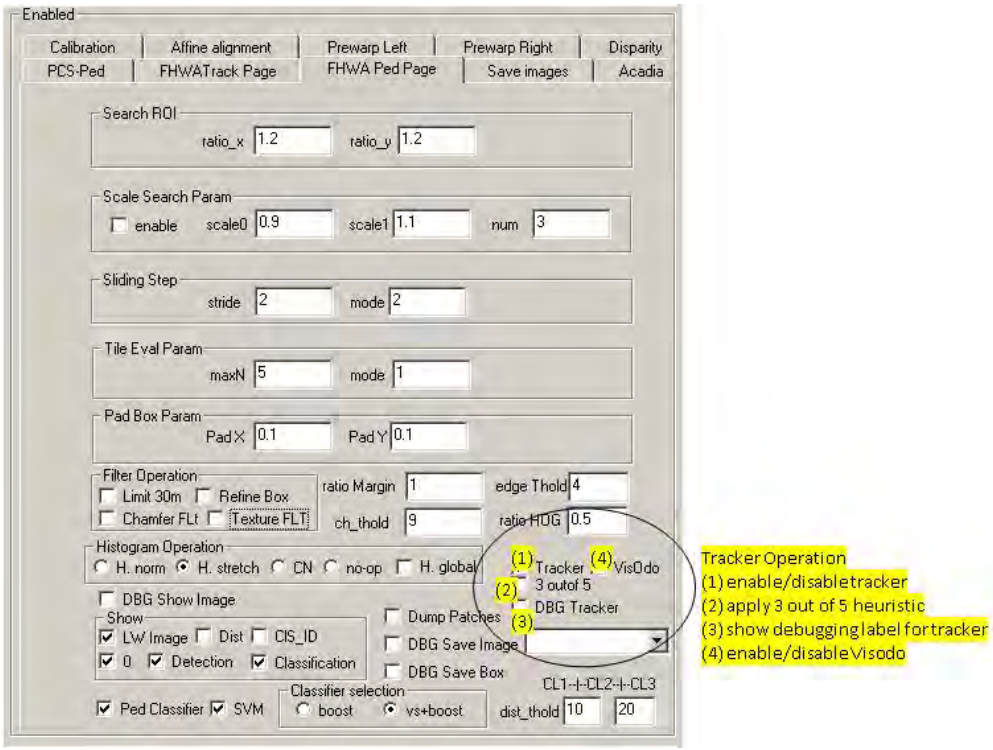


Figure 106. Screenshot. PC interface showing tracker options.

The boxes circled in figure 107 show the following classifier threshold options: (1) multiplier for the classifier threshold (large value suppresses detection), (2) edge detect threshold, (3) HOG feature threshold, and (4) threshold for chamfer filter.

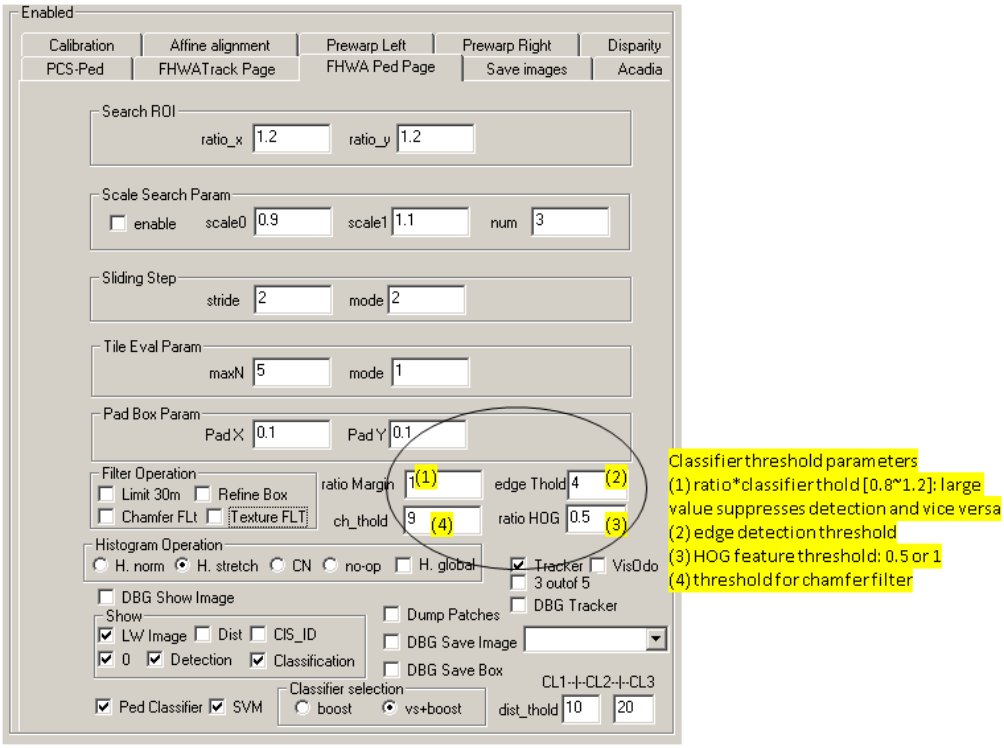


Figure 107. Screenshot. PC interface showing classifier threshold options.

The boxes circled in figure 108 indicate the options to set the tracker search range as a multiplier to the detection box position in the X and Y directions.

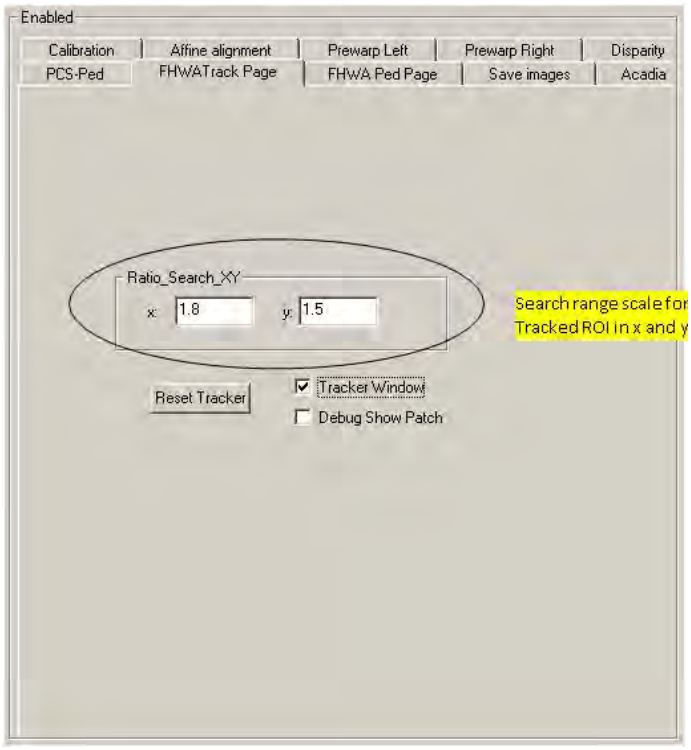


Figure 108. Screenshot. Pedestrian tracker interface showing options to set tracker search range.

The options circled in figure 109 indicate the following options: (1) reset the tracker, (2) display the tracker window, and (3) display the tracked patches for debugging purposes.

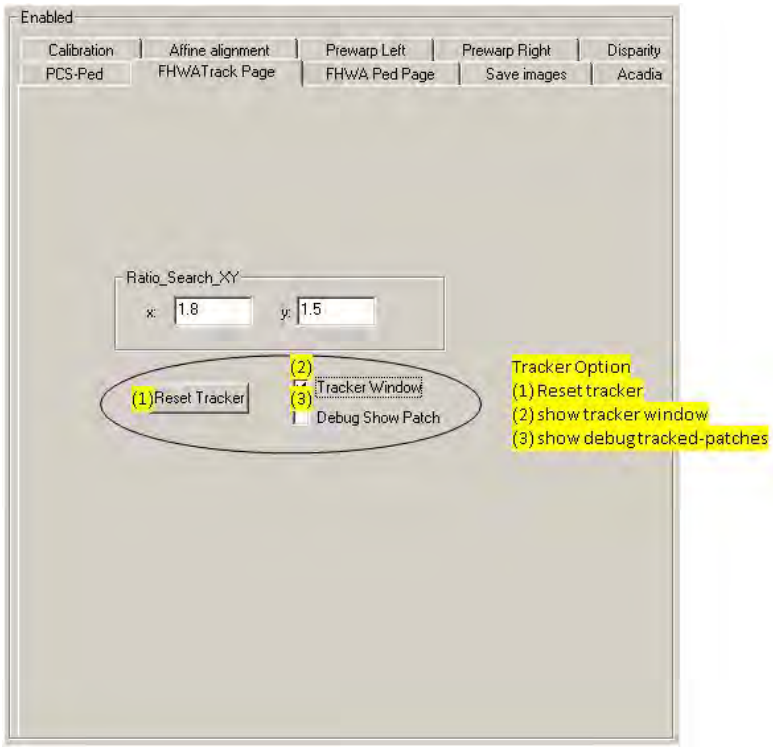


Figure 109. Screenshot. Pedestrian tracker interface showing tracker options.

ACKNOWLEDGEMENTS

The researchers of this project would like to express their appreciation to FHWA and National Highway Transportation Safety Administration staff members for their input and feedback over the course of the program. The interactions with the project panel were fruitful and helped set the course for the development of the project.

The images showing "©INRIA" belong to the Institut National de Recherche en Informatique et en Automatique (National Institute for Research in Computer Science and Control). Those images are screenshots from videos owned by INRIA. Some of the screenshots obtained from the INRIA videos contain an image overlay, such as boxes or colored areas. Those overlays were created specifically for this report and are not contained in the original videos owned by INRIA. The figure captions underneath the images from INRIA were written specifically for this report and do not appear in the original videos.

FHWA is using screenshots in this report from the videos owned by INRIA under a creative commons license located at http://www-sop.inria.fr/orion/ETISEO/iso_album/ETISEO_ UserAgreement.doc. Because FHWA does not own the graphic images, the agency cannot provide reprint or reuse permission. For reprint or reuse purposes, the original videos are located at http://pascal.inrialpes.fr/data/human/.

REFERENCES

1. National Highway Traffic Safety Administration. *Fatality Analysis Reporting System (FARS)*, Washington, DC. Obtained from: http://www.nhtsa.gov/FARS.

2. Insurance Institute for Highway Safety. *Fatality Facts 2007: Pedestrians*, Highway Loss Data Institute, Arlington, VA. Obtained from: http://www.iihs.org/research/fatality_ facts_2007/pedestrians.html. Site last accessed July 15, 2009.

3. Freund, Y. and Schapire, R.E. (1997). "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *Journal of Computer and System Sciences*, *55*(1), 119–139.

4. Friedman, J., Hastie, T., and Tibshirani, R. (2000). "Additive Logistic Regression: A Statistical View of Boosting," *The Annals of Statistics*, *28*(2), 337–407.

5. Kindermann, R. and Snell, J.L. (1980). *Markov Random Fields and Their Applications*, American Mathematical Society, Providence, RI.

6. Dalal, N. and Triggs, B. (2005). *Histograms of Oriented Gradients for Human Detection*, 886–893, Institute of Electrical and Electronics Engineers Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA.

7. Lowe, D.G. (2004). "Distinctive Image Features From Scale-Invariant Keypoints," *International Journal of Computer Vision*, *60*, 91–110.

8. Mohan, A., Papageorgiou, C., and Poggio, T. (2001). "Example-Based Object Detection in Images by Components," *Institute of Electrical and Electronics Engineers Transactions on Pattern Analysis and Machine Intelligence*, *23*, 349–361.

9. Ke, Y. and Sukthankar, R. (2004). *PCA-SIFT: A More Distinctive Representation for Local Image Descriptors*, 506–513, 2004 Institute of Electrical and Electronics Engineers Computer Society Conference on Computer Vision and Pattern Recognition, Washington, DC.

10. Tuzel, O., Porikli, F., and Meer, P. (2007). *Human Detection Via Classification on Riemannian Manifolds*, 2007 Institute of Electrical and Electronics Engineers Computer Society Conference on Computer Vision and Pattern Recognition, Minneapolis, MN.

11. Tran, D. and Forsyth, D.A. (2007). *Configuration Estimates Improve Pedestrian Finding*, 21st Annual Conference on Neural Information Processing Systems, Vancouver, Canada.

12. National Institute for Research in Computer Science and Control. *INRIA Person Dataset*. Normal Web site download from the INRIA organization: http://pascal.inrialpes.fr/ data/human/. See Acknowledgements section for reuse information.

13. Wu B. and Nevatia R. (2007). *Cluster Boosted Tree Classifier for Multi-View, Multi-Pose Object Detection*, Eleventh IEEE International Conference on Computer Vision, Rio de Janeiro, Brazil.

14. Leibe, B., Cornelis, N., and Cornelis, L.V.G.K. (2007). *Dynamic 3d Scene Analysis from a Moving Vehicle*, 2007 Institute of Electrical and Electronics Engineers Computer Society Conference on Computer Vision and Pattern Recognition, Minneapolis, MN.

15. Mikolajczyk, K. and Schmid, C. (2005). "A Performance Evaluation of Local Descriptors," *Institute of Electrical and Electronics Engineers Transactions on Pattern Analysis and Machine Intelligence, 27*(10), 1615–1630.

16. Gavrila, D.M. and Munder, S. (2007). "Multi-Cue Pedestrian Detection and Tracking From a Moving Vehicle," *International Journal of Computer Vision, 73*, 41–59.

17. Doll´ar, P., Wojek, C., Schiele, B., and Perona, P. (2009). *Pedestrian Detection: A Benchmark*, 2009 Institute of Electrical and Electronics Engineers Computer Society Conference on Computer Vision and Pattern Recognition, Miami, FL.

18. Shashua, A., Gdalyahu, Y., and Hayun, G. (2004). *Pedestrian Detection for Driver Assistance Systems: Single-Frame Classification and System-Level Performance*, Proceedings from the 2004 Institute of Electrical and Electronics Engineers Intelligent Vehicle Symposium, Parma, Italy.

19. Hoiem, D., Efros, A.A., and Hebert, M. (2005). *Geometric Context From a Single Image*, Institute of Electrical and Electronics Engineers Tenth International Conference on Computer Vision, Beijing, China.

20. Hoiem, D., Efros, A.A., and Hebert, M. (2006). *Putting Objects in Perspective*, Institute of Electrical and Electronics Engineers Computer Society Conference on Computer Vision and Pattern Recognition, New York, NY.

21. Wojek, C. and Schiele, B. (2008). *A Dynamic Conditional Random Field Model for Joint Labeling of Object and Scene Classes*, 10th European Conference on Computer Vision, Marseille, France.

22. Brostow, G., Shotton, J., Fauqueur, J., and Cipolla, R. (2008). *Segmentation and Recognition Using Structure from Motion Point Clouds*, 10th European Conference on Computer Vision, Marseille, France.

23. SRI International. ACADIA Video Processors. Obtained from: http://www.sarnoff.com/products/acadia-video-processors.

24. Burt, P. and Adelson, T. (1983). "The Laplacian Pyramid as a Compact Image Code," *Institute of Electrical and Electronics Engineers Transactions on Communications, 9*, 532–540.

25. Chang, P., Camus, T., and Mandelbaum, R. (2004). *Stereo-Based Vision System for Automotive Imminent Collision Detection, 2*004 Institute of Electrical and Electronics Engineers Intelligent Vehicles Symposium, 274–279, Parma, Italy.

26. Comaniciu, D. and Meer, P. (2002). "Mean Shift: A Robust Approach Toward Feature Space Analysis," *Institute of Electrical and Electronics Engineers Transactions on Pattern Analysis and Machine Intelligence, 24*, 603–619.

27. Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, First Edition, Springer, New York, NY.

28. Besag, J. (1974). "Spatial Interaction and the Statistical Analysis of Lattice Systems," *Journal of the Royal Statistical Society*, Series B (Methodological), *36*(2), 192–236.

29. Weiss, Y. and Freeman, W.T. (2001). "On the Optimality of Solutions of the Max-Product Belief Propagation Algorithm in Arbitrary Graphs," *Institute of Electrical and Electronics Engineers Transactions on Information Theory, 47.*

30. *Robust Multi-Person Tracking from Mobile Platforms*. Obtained from: http://www.vision.ee.ethz.ch/~aess/dataset/.

31. Ess, A., Leibe, E., Schindle, K., and Van Gool, L. (2009). "Robust Multi-Person Tracking from a Mobile Platform," *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Obtained from: ftp://ftp.vision.ee.ethz.ch/publications/articles/eth_biwi_00623.pdf.

INDEX

HRDS-10/10-12(WEB)E