3-10-2010

Improved Multispectral Skin Detection and its Application to Search Space Reduction for Dismount Detection Based on Histograms of Oriented Gradients

Adam L. Brooks

Follow this and additional works at: https://scholar.afit.edu/etd

Part of the Electrical and Electronics Commons, and the Systems and Communications Commons

Improved Multispectral Skin Detection and its
Application to Search Space Reduction for
Dismount Detection Based on
Histograms of Oriented Gradients

THESIS

Adam Lee Brooks, Captain, USAF

AFIT/GE/ENG/10-05

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GE/ENG/10-05

IMPROVED MULTISPECTRAL SKIN DETECTION AND ITS
APPLICATION TO SEARCH SPACE REDUCTION FOR
DISMOUNT DETECTION BASED ON
HISTOGRAMS OF ORIENTED GRADIENTS

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

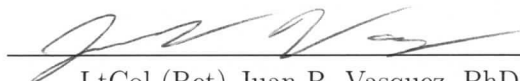Adam Lee Brooks, BS

Captain, USAF

March 2010

AFIT/GE/ENG/10-05

IMPROVED MULTISPECTRAL SKIN DETECTION AND ITS
APPLICATION TO SEARCH SPACE REDUCTION FOR
DISMOUNT DETECTION BASED ON
HISTOGRAMS OF ORIENTED GRADIENTS

Adam Lee Brooks, BS
Captain, USAF

Approved:

_____   _____
Maj Michael J. Mendenhall, PhD   5-MAR-2010
(Chairman)   date

_____   _____
LtCol (Ret) Juan R. Vasquez, PhD   5 Mar 2010
(Member)   date

_____   _____
Dr Gilbert L. Peterson   5 MAR 2010
(Member)   date

AFIT/GE/ENG/10-05

Abstract

Due to the general shift from conventional warfare to terrorism and urban warfare by enemies of the United States in the late 20th Century, locating and tracking individuals of interest have become critically important. Dismount detection and tracking are vital to provide security and intelligence in both combat and homeland defense scenarios including base defense, combat search and rescue (CSAR), and border patrol.

This thesis focuses on exploiting recent advances in skin detection research to reliably detect dismounts in a scene. To this end, a signal-plus-noise model is developed to map modeled skin spectra to the imaging response of an arbitrary sensor, enabling an in-depth exploration of multispectral features as they are encountered in the real world for improved skin detection. Knowledge of skin locations within an image is exploited to cue a robust dismount detection algorithm, significantly improving dismount detection performance and efficiency.

This research explores multiple spectral features and detection algorithms to find the best features and algorithms for detecting skin in multispectral visible and short wave infrared (SWIR) imagery. This study concludes that using SWIR imagery for skin detection and color information for false alarm suppression results in 95% probability of skin detection at a false alarm rate of only 0.4%.

Skin detections are utilized to cue a dismount detector based on histograms of oriented gradients. This technique reduces the search space by nearly 3 orders of magnitude compared to searching an entire image, while reducing the average number of false positives per image by nearly 2 orders of magnitude at 95% probability of dismount detection. The skin-detection-cued dismount detector developed in this thesis has the potential to make significant contribution to the United States Air Force human measurement and signature intelligence and CSAR missions.

iv

Acknowledgements

Table of Contents

List of Figures
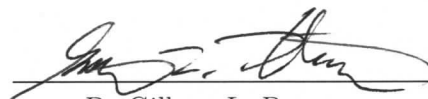
xiii

xiv

List of Tables

Improved Multispectral Skin Detection and its Application to Search Space Reduction for Dismount Detection Based on Histograms of Oriented Gradients

I. Introduction

The United States Air Force (USAF) has made intelligence, surveillance, and reconnaissance (ISR) capabilities a high priority. The Air Force Doctrine Document 1 (AFDD-1) states that "As a leader in the military application of air, space, and intelligence, surveillance, and reconnaissance technology, the Air Force is committed to innovation to guide research, development, and fielding of unsurpassed capabilities" [5].

Due to the general shift from conventional warfare to terrorism and urban warfare by enemies of the United States in the late 20th Century, locating and tracking individuals of interest has become of vital importance [5]. Several research efforts address this growing need for human surveillance and tracking including

- The 2003 Defense Advanced Research Projects Agency (DARPA) Combat Zones that See (CTS) program [4], [29] which has the goal of creating a dense network of inexpensive cameras and sensors to monitor "everything that moves" on a full-city scale and report all observations to a central operating center. The research was meant to be applied to an urban combat zone to help protect soldiers on the ground by improving battlefield awareness.

- The United States Army funded the development of algorithms for unmanned air vehicle (UAV) ISR systems for tracking targets in urban environments as part of the Army's 2007 Small Business Technology Transfer Program [70], [73]. Targets of interest included humans, civilian vehicles, and military targets that may exhibit highly nonlinear motions.

Dismount detection is the critical first step to successful dismount tracking. The overarching goal of this thesis effort is to leverage multispectral skin detection to augment a state-of-the-art dismount detection methodology.

1.1 Problem Statement

Modern shape-based dismount detection techniques are often either computationally expensive due to the size of the search space or application-limited due to constraints imposed by search space reduction techniques. Shape-based detectors also tend to have a high confusion rate with human-like objects in a scene. Examples of common false alarm sources for shape-based detectors include parking meters, signs, small trees, fire hydrants, or anything with similar vertical structure [25].

The goal of this research is to provide a robust method of reducing the search space for a modern shape-based dismount detector using multispectral skin detections as cueing sources. Additionally, it is hypothesized that using skin detections for cueing a shape-based dismount detector will significantly reduce false alarms attributed to human-like objects. Since typical urban false alarm sources are unlikely to have material properties similar to exposed skin, skin detection cueing will likely reject many common false alarm sources from the search space.

1.2 Scope

The scope of this thesis effort must be limited in order to accomplish the research goals mentioned above. To that end, the tasks accomplished by this effort are as follows:

1. Develop a signal-plus-noise model to map modeled skin spectra to the imaging response of an arbitrary sensor.

2. Compare the performance of multiple spectral features for suppressing false alarms in skin detection using both modeled and real-world data.

3. Compare the performance of multiple skin detection algorithms using both modeled and real-world data.

4. Develop a method for using skin detections to cue a dismount detector.

5. Compare the performance of one existing sliding-window dismount detector with a skin detection-cued version of the same detector using multispectral data.

The signal-plus-noise model is developed by adding sensor noise components that are experimentally determined for a sensor of interest. Specular reflection components are added until the modeled data are visually similar to skin data collected by the imager. The signal-plus-noise model is presented in Section 2.7.7 and Section 4.3.

The normalized difference vegetation index (NDVI) and normalized difference green-red index (NDGRI) skin spectral features (presented in Section 2.7.3) are compared in terms of false alarm suppression performance for the skin detection algorithms implemented in this thesis effort. Rules-based and likelihood ratio test (LRT)-based skin detection algorithms are presented in Section 2.7.8 and Section 3.2.3 respectively, while comparisons of skin detection performance between spectral features and between algorithms are presented in Section 4.2.

Methods and considerations for using skin detections to cue a dismount detector are discussed in Section 3.3. Only the dismount detector based on histograms of oriented gradients (HOG) is tested for comparison. A recent effort in [25] compares the performance of several state-of-the-art dismount detectors. The end result of the work in [25] showed that the HOG-based sliding-window dismount detector outperformed the other methods researched, making in-depth comparison of those detection techniques unnecessary for the purposes of this effort. Performance comparison results are presented in Section 4.5.

1.3   Document Organization

Chapter II of this document provides the necessary background information for this thesis. This background information describes the basic tracking framework,

dismount detection techniques, the properties of human skin, and the signal processing and classification techniques used throughout this thesis effort.

Chapter III provides the methodology employed for this effort. Included in this discussion are a skin detection algorithm based on a likelihood-ratio test (LRT) and methodology for using skin detections to cue search windows for a HOG-based dismount detection system.

Chapter IV provides experimental results and analyses of the results. Included in this discussion are data set descriptions; designs of experiments; and performance comparisons for skin detector features, skin detectors, and dismount detectors.

Chapter V provides conclusions drawn from the analyses of results mentioned in Chapter IV. Specifically, Chapter V includes a summary of results, list of contributions this research effort provides, and recommendations for future work.

Appendix A presents the basics of bilinear interpolation. Appendix B presents the skin detection masks and skin-detection-cued HOG-based dismount detections for each HyperSpecTIR version 3 (HST3) image used in this thesis effort. Appendix C is an electronic appendix ("AppendixC.pdf" on the included disc) that lists the full set of experimentally-derived expectation maximization (EM) -Gaussian mixture model (GMM) parameters determined by this thesis effort for LRT-based skin detection.

II. Background

This chapter provides an overview of how detection systems fit into a tracking framework, how others have approached the problem of dismount detection, and general background information on the hyperspectral properties of human skin. Most importantly, this chapter provides the essential background needed for spatial features, spectral features, image processing techniques, classifier architectures, and detection algorithms that are implemented in this thesis effort.

The chapter begins with an overview of how detectors fit into a tracking architecture. Next is an overview of passive sensors often used for tracking. Next is a review of current state-of-the-art techniques used for detecting dismounts, followed by in-depth descriptions of the spatial feature and detector that is implemented directly from that research for the purposes of this thesis effort.

An overview of the sliding-window detection scheme and its search-space limitations is provided, followed by common techniques for sliding-window-detector search-space reduction and their limitations. Next, an overview of hyperspectral image processing and the hyperspectral properties of human skin, which are exploited by this thesis effort to aid sliding-window search-space reduction is provided.

The final portion of this chapter provides methodology for approximating the functional form of a probability density function for incomplete data and applying that approximation to the likelihood ratio test, a detector scheme that minimizes the Bayes risk.

2.1 Notation and Terminology Conventions

Due to the large number of variables and parameters that are used in this thesis effort, some common naming conventions are established for consistency and readability. All letter assignments as variables in this section are strictly for demonstration purposes only.

2.1.1 Underline and Boldface Notation.   Underline notation is used to differentiate between scalars, vectors, two-dimensional matrices (henceforth *matrices*), and three-dimensional matrices (henceforth *cubes*).

Lowercase variables that have no special typeface and no underline are considered scalars (e.g., s is a scalar). Variables that have a single underline are considered vectors (e.g., $\underline{v}$ is a vector). Variables that have a double underline are considered matrices (e.g., $\underline{\underline{M}}$ is a matrix). Variables that have a triple underline are considered cubes (e.g., $\underline{\underline{\underline{C}}}$ is a cube).

Boldface notation is used to indicate that a variable is a *structure* (e.g., **S** is a structure). Structures are used when data do not fit into the scalar, vector, matrix, or cube paradigm. Structures are often used to organize several disparate forms of information that are associated with one another (e.g., a string with the file name, an arbitrary number of image patches, and class labels associated with those image patches).

2.1.2 Subscript Notation.   Subscripts are typically used to indicate that a variable is the subscripted element of a higher-dimensional set. For example, vectors are defined as a set of scalars, so v_i is the ith element of the vector $\underline{v}$. Multiple levels are transcended by multiple subscripts (e.g., $m_{i,j}$ is the ith element of vector $\underline{m}_j$, which is the jth vector of matrix $\underline{\underline{m}}$).

The length of each subscripted dimension is defined at the time that the variable is defined (e.g., $x_m, m \in \mathbb{Z}[1, M]$ indicates that the subscript m can have any integer value from 1 through M inclusive and that $\underline{x}$ is of length M).

2.1.3 Special Subscripting Cases.   Some subscripted variables do not imply that they are an element of a larger set. Those cases are specifically defined at the time of use. For example, subscripted decision spaces S_i are used to define the class i that a sample will be labeled by a detector.

Some subscripts are only meant to indicate global conventions that are used for different purposes. For example, η is reserved to indicate thresholds. Thresholds for different algorithms are subscripted based on the algorithm they apply to (e.g., η_Ω is a threshold on Ω, η_Λ is a threshold on Λ).

2.1.4   Inner Product Notation.   The *inner product* or *dot product* of two equal-length vectors ($\underline{a}$ and $\underline{b}$) is notated as

$$\langle \underline{a}, \underline{b} \rangle = \sum_{n=1}^{N} a_n b_n, \tag{2.1}$$

where N is the length of $\underline{a}$ and $\underline{b}$.

2.1.5   Variants of the Same Variable.   Above-letter symbols are used to differentiate between different versions of the same base variable. Hat notation is used to indicate that a variable obtains its value from estimation or approximation of the base variable's true value (e.g., $\hat{e}$ is an estimate of the variable e).

Tilde notation is used to indicate that a variable obtains its value from a model of the base variable (e.g., $\tilde{m}$ is a modeled version of the variable m).

Prime notation is used to indicate that a variable may have undergone an optional process, therefore the variable's value may be that of the original base variable or modified by the optional process (e.g., o' can either be the original value o or a processed version of o).

Dot notation is used to indicate that a variable is the derivative of the base variable (e.g., $\dot{d}$ is the first derivative of d).

2.1.6   Detector Terminology.   For the purposes of this thesis, a set of common detector terminology is defined for consistency. A *window* is defined as a two-dimensional bounding box within an image. A *search window* or *detector window* is

2-3

defined as a window that is to be evaluated by a detector to determine the class of the contents of that window.

An *alarm* is defined as a sample that a detector decides is in the class of interest (the *positive class*). An alarm is synonymous with a *detection* from other common detector terminology. An *alarm window* is defined as a search window whose contents a detector decides are in the positive class. A *rejection* is a sample that the detector decides is outside the class of interest, or in the *negative class*.

A *hit* is defined as an alarm that is truly in the positive class (i.e., a correct positive decision). A *false alarm* is defined as an alarm that is truly in the negative class (i.e., an incorrect positive decision). A *correct rejection* is defined as a rejection that is truly in the negative class (i.e., a correct negative decision). A *miss* is defined as a rejection that is truly in the positive class (i.e., an incorrect negative decision).

The space that contains all possible observations is defined as $\mathcal{S}$. For a binary detector, $\mathcal{S}$ is partitioned into two *decision regions* as

$$S_i : i = \begin{cases} 1 & \text{if criteria for } i = 1 \text{ are met} \\ 0/-1 & \text{if criteria for } i = 0/-1 \text{ are met} \end{cases}, \tag{2.2}$$

where S_i is the decision region where the class label i is assigned to an evaluated sample, $S_1 \cup S_{0/-1} = \mathcal{S}$ and $S_1 \cap S_{0/-1} = \varnothing$. Equation (2.2) is an example of a *decision rule*. All detectors described in this thesis employ a decision rule in a format similar to Eqn. (2.2). For all detectors in this thesis, S_1 is the decision region for the positive class and S_0 (or S_{-1} depending on the algorithm) is the decision region for the negative class.

2.2 *Tracking Architecture*

Figure 2.1 illustrates the basic structure of a hyperspectral or multispectral-based tracking architecture [11], [69], [70]. First, raw imagery are passed from the imaging system to a detector (a dismount detector for this thesis effort). The detector
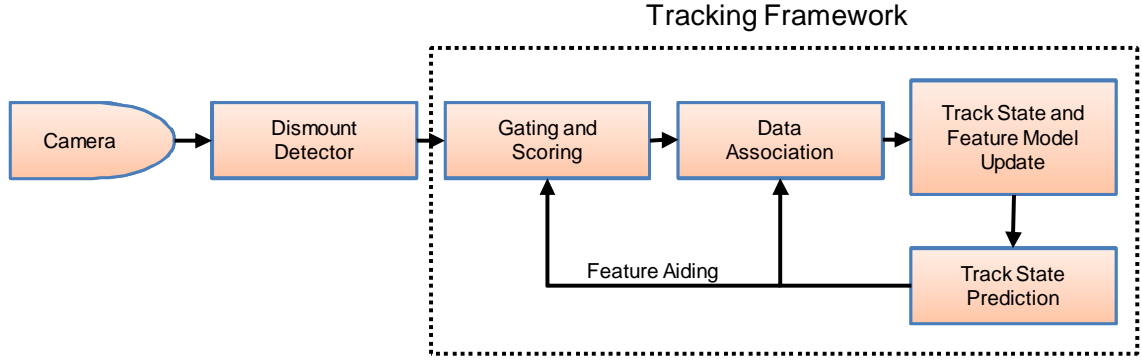
Figure 2.1:   Dismount tracking taxonomy.

finds objects of interest within the imagery and passes information about the location and identity of objects of interest to the tracker portion of the architecture. From the diagram in Fig. 2.1, it is clear that detector performance has a significant impact on overall tracking performance since the tracker relies on data provided by the detector.

In a feature-aided tracker, spatial, spectral and other information about detected targets is used to augment track association beyond the typical kinematics-only approach. Since the dismount detector described in this thesis requires multispectral information and generates highly-descriptive spatial and spectral features, it is possible that those constituent data may be useful for feature-aided tracking. This thesis effort does not focus on feature-aided tracking, however, since it is outside the scope outlined in Section 1.2.

2.3   Passive Sensors Used for Tracking

Several types of passive sensors are used for tracking dismounts. Cameras sensitive to the visible region of the electromagnetic spectrum are the most common due to their low cost and high image quality. Both monochrome and red-green-blue (RGB) visible cameras are frequently used. Generally, these cameras are advantageous for generating spatial features for detecting specific target classes [18], [19], [65], [66], [78], [80]. Additionally, RGB cameras (or cameras using similar three-

channel color spaces [7], [28]) can be used to generate spectral features for skin detection [12], [26], [32], [59], [74].

Infrared cameras are used less frequently than visible cameras, mainly due to expense and comparatively poor image quality. Cameras sensitive to the mid-wave infrared (MWIR) and long-wave infrared (LWIR) regions of the electromagnetic spectrum (3000-5000nm and 8000-12000nm or 7000-14000nm respectively [44], [45]) are often utilized because they are sensitive to thermal emissions and can therefore detect body heat. They can be very effective in certain environments at detecting thermal signatures. However, advances in thermal-masking clothing could limit their potential use in military applications. Additionally, these systems may be less effective in recovery missions due to the reduced thermal signature of a corpse. Poor contrast may also be an issue in climates near body temperature.

Cameras sensitive in the near-infrared (NIR) and short-wave infrared (SWIR) regions of the spectrum (700-1000nm and 1000-3000nm respectively [44], [45]) are less widely-used due to high cost and limited applications. They do not share the image quality and resolution benefits of visible sensors, nor do they have the ability to detect thermal signatures as do sensors sensitive in the MWIR and LWIR. They are most commonly used for very specific applications that require information from the SWIR region of the electromagnetic spectrum. Specific applications that utilize NIR and SWIR imagery include skin detection [35], counting vehicle occupants [58], and face detection [20].

Hyperspectral cameras are most commonly used for geographical survey and remote sensing applications. Typically, these are line-scanning cameras that are sensitive to hundreds of narrow regions of the electromagnetic spectrum, nominally ranging from 400-2500nm. As such, they often have very low frame rates and spatial resolution. Additionally, the large amount of data they collect per frame requires extensive computational power to process. The advantage of hyperspectral cameras is for feature-aided tracking [11], [69], [70]. Due to the richness of spectral data available,

highly discriminative spectral features can often be generated for detecting specific target classes [40].

Multi-spectral camera systems are often developed to detect specific wavelengths of interest for a particular detection task. Often they are a combination of multiple cameras sensitive in the broad range of desired wavelengths and filtered at the specific wavelengths of interest. This scheme provides many of the benefits of hyperspectral imaging for detecting spectral features, while significantly reducing the amount of data collected and therefore lowering computational expense. Additionally, since line-scanning cameras are often not required for the few wavelengths needed, frame rates and resolution can be improved dramatically over those of line-scanning hyperspectral cameras. Specific applications for multispectral sensors include background modeling and object tracking [14], [16].

2.4 State-of-the-art Dismount Detection Techniques

There are numerous approaches to the problem of dismount detection. The most common approach to dismount detection is the whole-body detection approach. In this approach, a classifier is trained based on a set of exemplars or codebook patches.

Spatial features of an object are often utilized to increase separability of object classes. These features include, but are not limited to, nonadaptive Haar wavelet features [46], [57], [67], [75], dense encoding of local edge orientations (i.e., HOG) [18], [19], [65], [66], [78], [80], and sparse encoding of local edge orientations (i.e., scale-invariant feature transform (SIFT)) [39]. One challenge for the whole-body approach is the number of exemplars necessary to represent the full diversity of pose configurations within the classifier training set.

Another approach to dismount detection combines expert body part detectors in an attempt to assemble a stronger "ontological" representation of a dismount [27], [49], [65], [67], [76], [78]. This approach often breaks the body down into combinations

2-7

of subpart detectors (i.e., torso, legs, arms, and head) [8], [43], [46], [65], [68], [76] or a codebook representation [6], [37], [38], [64].

One challenge for the ontological approach is associating multiple subpart detections together to determine the likelihood that a dismount is present. One solution is to train a combination classifier [8], [46], [65]. Probabilistic inference of the most likely object configuration observed is another solution to the problem of associating multiple parts detectors in a meaningful way [43], [68], [76].

A more exhaustive survey of state-of-the-art dismount detection techniques is provided by [25]. This thesis effort focuses on the full-body detection approach using HOG features combined with linear support vector machines (linSVM) .

2.5   Histograms of Oriented Gradients-based Dismount Detection

This section provides the background necessary to construct the basic components of a HOG-based dismount detector. First, the methodology for generating HOG features is provided, followed by a description of how a linSVM works. Finally, the bootstrapping technique for training discrimination-based classifiers is provided.

2.5.1   Histograms of Oriented Gradients Feature Generation.   One of the most popular spatial features used in current literature is the HOG feature [19], [25], [65], [66], [78], [80]. The feature is commonly used in concert with a sliding-window detector for detecting and classifying in-scene objects. For the purposes of this thesis, only the HOG parameter set that performed best in [25] is discussed and implemented. Exploration of the best HOG parameters to use for dismount detection is beyond the scope of this thesis, especially since that study is specifically accomplished in [25].

Figure 2.2 illustrates the steps involved in HOG feature generation. First, an image patch is scaled to a resolution of 48×96 pixels (leaving a 12-pixel border around dismounts for training purposes). Next, the image gradient is calculated by convolving the image with a $(-1,\ 0,\ 1)$ mask without smoothing in both the x and y-directions. Figure 2.3 illustrates how this convolution affects imagery. Consider a
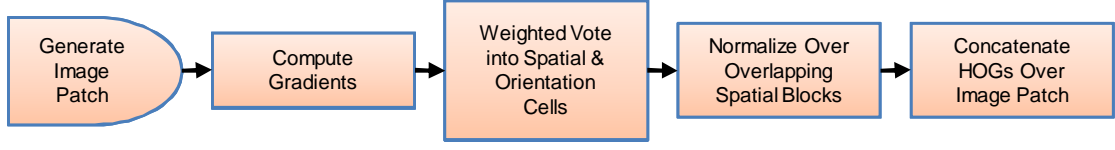
Figure 2.2:   Histograms of oriented gradients (HOG) feature generation process (inspired by Fig. 1 in [19]).

row of pixels with values as indicated in the top portion of Fig. 2.3. Note the high-contrast transitions in pixel values highlighted in blue, red, magenta and green in the top portion of Fig. 2.3. The bottom portion of Fig. 2.3 represents the *row gradient*, the result of convolving the top portion of Fig. 2.3 with the mask in the middle portion of Fig. 2.3. At each high-contrast transition point in the original image, there is a 2-pixel-wide impulse of magnitude equal to the change in pixel value in the original image. Directionality of the pixel value transition affects the sign of the gradient impulse.

Resulting x and y gradients (∇x and ∇y) are combined to produce gradient magnitude (r) and orientation ($\phi \in \mathbb{R}[0°, 180°]$) by

$$r = \sqrt{(\nabla x)^2 + (\nabla y)^2}, \tag{2.3}$$

$$\phi = \arctan \frac{\nabla y}{\nabla x}. \tag{2.4}$$

Gradient orientations are rotated by $\pm 180°$ as necessary to fall within $\mathbb{R}[0°, 180°]$ per the suggestion of [19].

Next, the image patch is subdivided into non-overlapping cells of 8×8 pixels, as depicted in Fig. 2.4 (red). For each cell, a 9-bin orientation histogram is taken (see Fig. 2.5). Each cell pixel contributes its gradient magnitude as a histogram vote. The magnitude is divided among the two bins whose centers are closest to the orientation of the pixel. The percentage of the vote that goes to each bin is determined by linear interpolation of the distance of the pixel orientation from each bin. The closer a bin
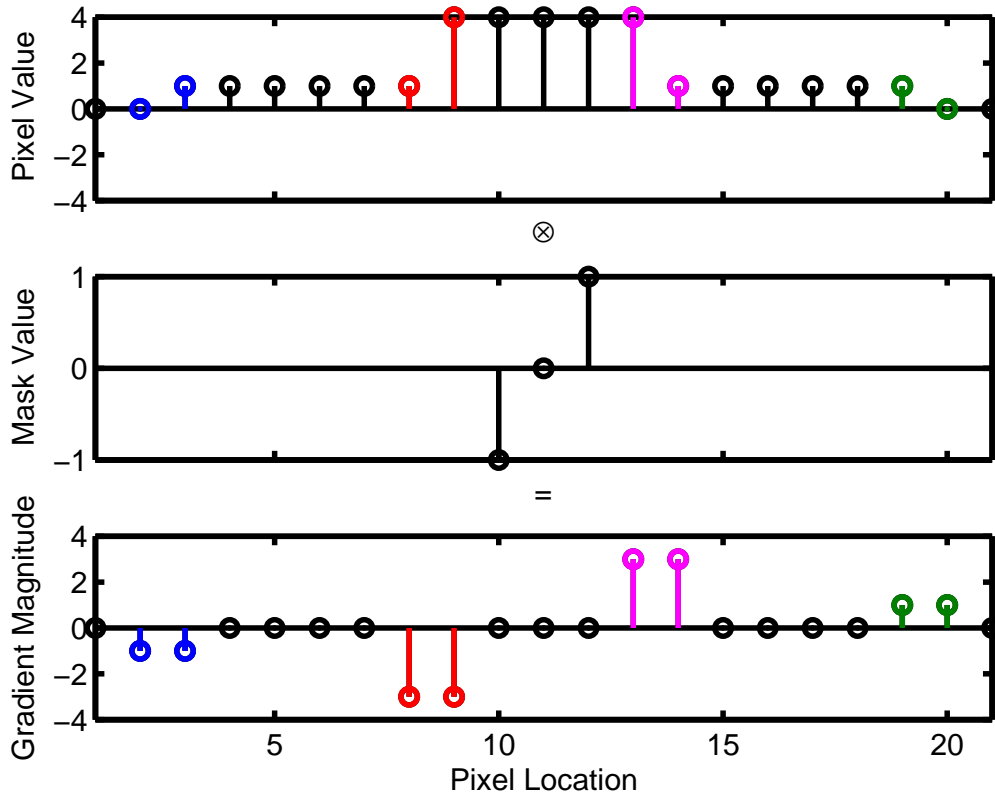
Figure 2.3: Gradient computation toy example. Blue, red, magenta, and green values represent locations of high-contrast pixel-value transitions in the original image.

center is to the pixel orientation, the greater percentage of the vote it receives. In the case that multi-channel imagery are used, the vote for each pixel is determined by the channel with the greatest gradient magnitude for that pixel.

Figure 2.5 depicts how a pixel contributes its vote to the histogram. If the pixel has a gradient magnitude of 100 units and orientation of 25° (black arrow), the bin centered at 30° will receive 75 units (blue arrow) and the bin centered at 10° will receive 25 units (red arrow) because the pixel orientation is 75% closer to the 30° bin center than the 10° bin center. This voting scheme is necessary to prevent aliasing. If the votes were simply quantized into the nearest bin, detailed orientation information would be destroyed. This histogram voting scheme incorporates all orientation information available, resulting in a more accurate representation of the cell.
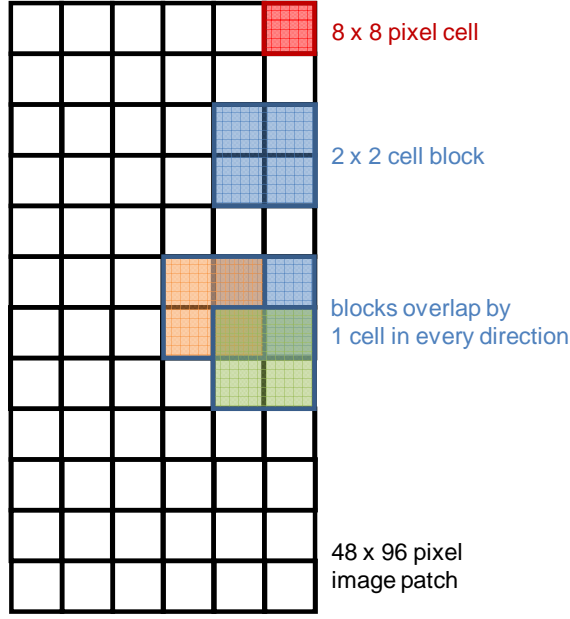
Figure 2.4: The image patch is subdivided into cells of 8×8 pixels (red) with no pixel overlapping of cells. Cells are grouped into blocks (blue) of 2×2 cells with an overlap of 1 cell in each direction (blue, orange, green).

Once histograms are calculated for each cell, the image patch is divided into blocks of 2×2 cells (Fig. 2.4 blue) with an overlap of one cell (Fig. 2.4 orange, green, and blue). For each block, the constituent cell histograms are concatenated together and the resulting vector is normalized by its ℓ_2-norm so that the vector has unit magnitude. The normalized vectors from each block in the image patch are finally concatenated together to form a 1980-dimensional HOG feature for a 48×96-pixel image patch[1]. In general, the length of the HOG feature is determined by

$$\text{length} = (\#\text{bins}) \times (\#\text{cells per block}) \times (\#\text{blocks}), \tag{2.5}$$

$$\#\text{blocks} = \left(\frac{w_x}{\#\text{pixels per cell}} - 1 \right) \times \left(\frac{w_y}{\#\text{pixels per cell}} - 1 \right). \tag{2.6}$$

---

[1]The 48×96-pixel image patch is suggested by [25] for dismount detection.
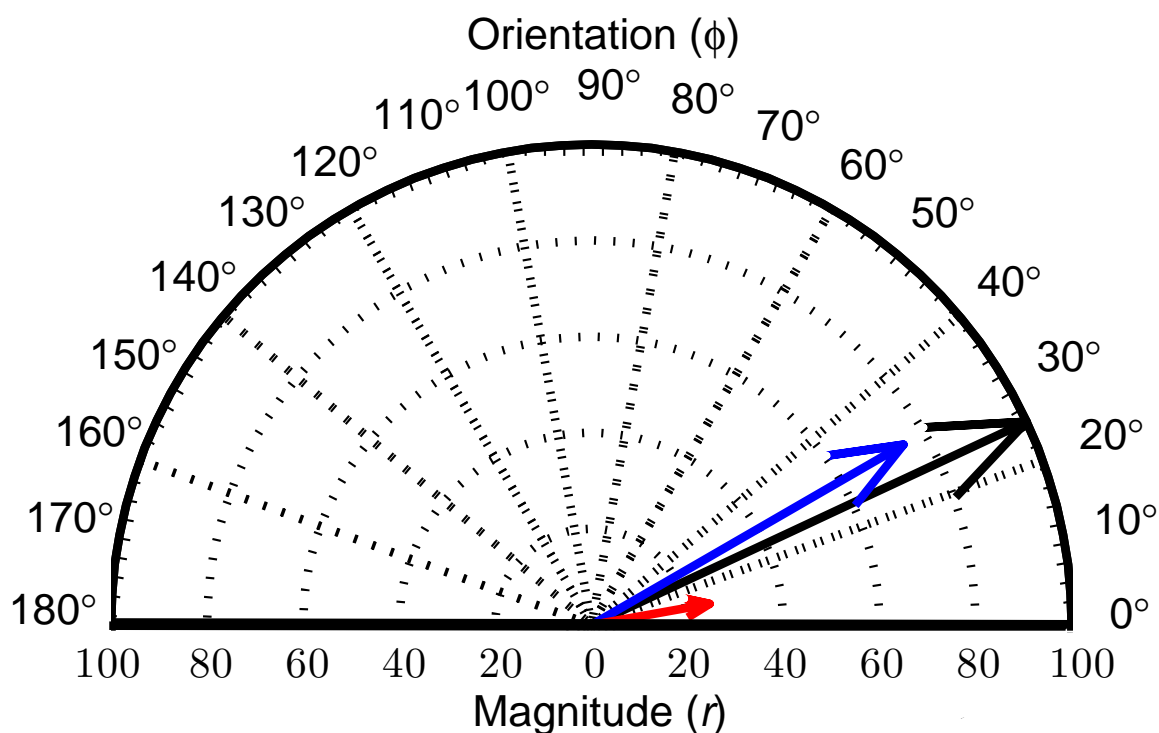
Figure 2.5:   Gradient orientation histogram voting toy example. Divisions along the ϕ-axis represent orientation histogram bin edges. Divisions along the r-axis are to aid visual interpretation of magnitude values. The black arrow represents a pixel gradient. The blue arrow represents the portion of the pixel gradient's magnitude that is received by the orientation bin centered at 30°. The red arrow represents the portion of the pixel gradient's magnitude that is received by the orientation bin centered at 10°.

2.5.2   Support Vector Machines.   There are several techniques for binary classification (deciding whether a sample is in a class or not in the class). One popular family of binary classification techniques is linSVMs [63].

Suppose a matrix of M, N-dimensional pattern vectors ($\underline{x}$) and a length-M vector of corresponding class labels ($\underline{y} \in \{\pm 1\}$) exist. Any N-dimensional hyperplane can be defined as follows:

$$\langle \underline{w}_m, \underline{x}_m \rangle + b = 0, \tag{2.7}$$

where $\underline{w}_m$ is a weight vector corresponding to pattern vector $\underline{x}_m$ and b is a real-valued offset. If the two classes are linearly-separable, a hyperplane can be defined to serve as a decision boundary between two classes as

$$S_i : i = \begin{cases} 1 & \text{if } \langle \underline{w}_m, \underline{x}_m \rangle + b > \eta_\tau \\ -1 & \text{if } \langle \underline{w}_m, \underline{x}_m \rangle + b < \eta_\tau \end{cases}, \tag{2.8}$$

where S_i is the decision space for class label i and η_s is a linSVM decision threshold that is typically set to 0, but can be varied to produce receiver operating characteristic (ROC) curves.

The *margin* is defined as the minimum distance from the decision boundary to any pattern vector as follows:

$$\varepsilon \equiv \min_m \frac{y_m \langle \underline{w}_m, \underline{x}_m \rangle}{\|\underline{w}_m\|}. \tag{2.9}$$

Figure 2.6 depicts examples of multiple possible separating hyperplanes for a two-class dataset and their associated margins.

Note that only the pattern vectors associated with the ε-value are necessary to define the hyperplane. This subset of the original pattern vectors ($\underline{\alpha} \subseteq \underline{x}$) is defined as the set of *support vectors*. The number of support vectors may be significantly smaller than M, eliminating the need to store the entire set of pattern vectors when using the linSVM on unknown data.

For optimal classification performance, the hyperplane with the largest margin should be chosen to serve as the decision boundary. If no hyperplane exists that perfectly separates the two classes, a *soft margin* optimization can be used. In soft-margin optimization, a cost is assigned to every mis-classified sample that is relative to the distance from the mis-classified sample to the decision hyperplane. The hyperplane with the largest margin and lowest cost is chosen to serve as the decision boundary [63].
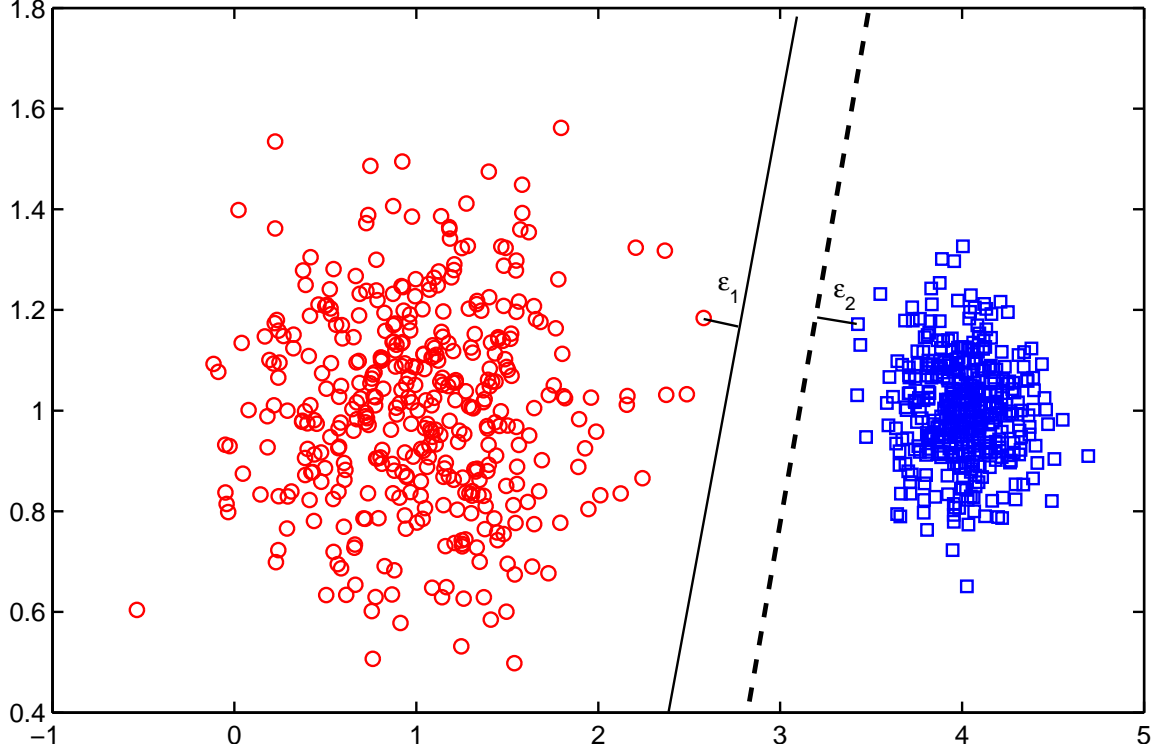
Figure 2.6:   Separating hyperplanes and margins toy example.

Determining the optimal values of $\underline{w}$ and b, and selecting $\underline{\alpha}$ corresponding to the optimal decision hyperplane is an optimization problem that is beyond the scope of this thesis. Details of how to solve the optimization problem (including cost parameter estimation and extension of SVM using kernel methods) are provided in [63]. An extensive list of freely available software implementations for learning and applying linSVMs can be obtained online [62].

2.5.3 Bootstrapping.   When training a classifier, it is important that each class is accurately represented in the training set. For a binary detector scenario, where the classifier simply distinguishes whether a sample is in the desired class or not, finding a useful training set can be tricky. For the positive training samples, often all that is needed is a representative group of samples from the positive class. The negative class, however, is defined as "everything else". Representing the entire universe outside the class of interest is impractical. Therefore, a technique known as *boot-*

strapping can be used to help define the most important aspect of any discriminative (decision boundary-based) classifier: the optimal decision boundary. The description of bootstrapping provided in this section is consistent with methods discussed in [19], [25], which should not be confused with the traditional definitions of bootstrapping (or bootstrap aggregating, "bagging") discussed in [10], [13], [21], [31], [56], [72].

Bootstrapping requires multiple classifier training steps. In the first step, the classifier is trained with an equal number of positive and negative samples. The negative samples are chosen at random from a large pool of known negative samples.

After the initial training step converges[2], the resulting classifier is used to classify additional random samples from the negative sample pool. The goal is to find as many false positives as there are positive training samples. Essentially, this step detects negative samples that are as close as possible to the best-performing decision boundary. Once hard false positives are identified, those false positives are added to the negative training set and the classifier is retrained.

Figure 2.7 illustrates the principles of bootstrapping using two-dimensional toy data. The blue squares represent known positive training samples. The red circles represent a random sub-sampling of known negative training samples. The red line represents an approximate maximum-margin decision boundary based on just the red and blue data. The black pluses represent false alarms when the red decision boundary is applied to another random sub-sampling of the known negative training pool. The black line represents a new decision boundary based on the false alarms from the red decision boundary. This is considered one bootstrapping step.

After each training iteration, the performance of the resulting classifier should be tested using a known test set. The test set should not include any of the training samples to avoid biasing the results. The bootstrapping process should continue to iterate until the classifier performance on the test set saturates based on user-defined saturation criteria, for example if the performance gain between iterations is

---

[2]Convergence criteria vary based on the type of classifier being trained.
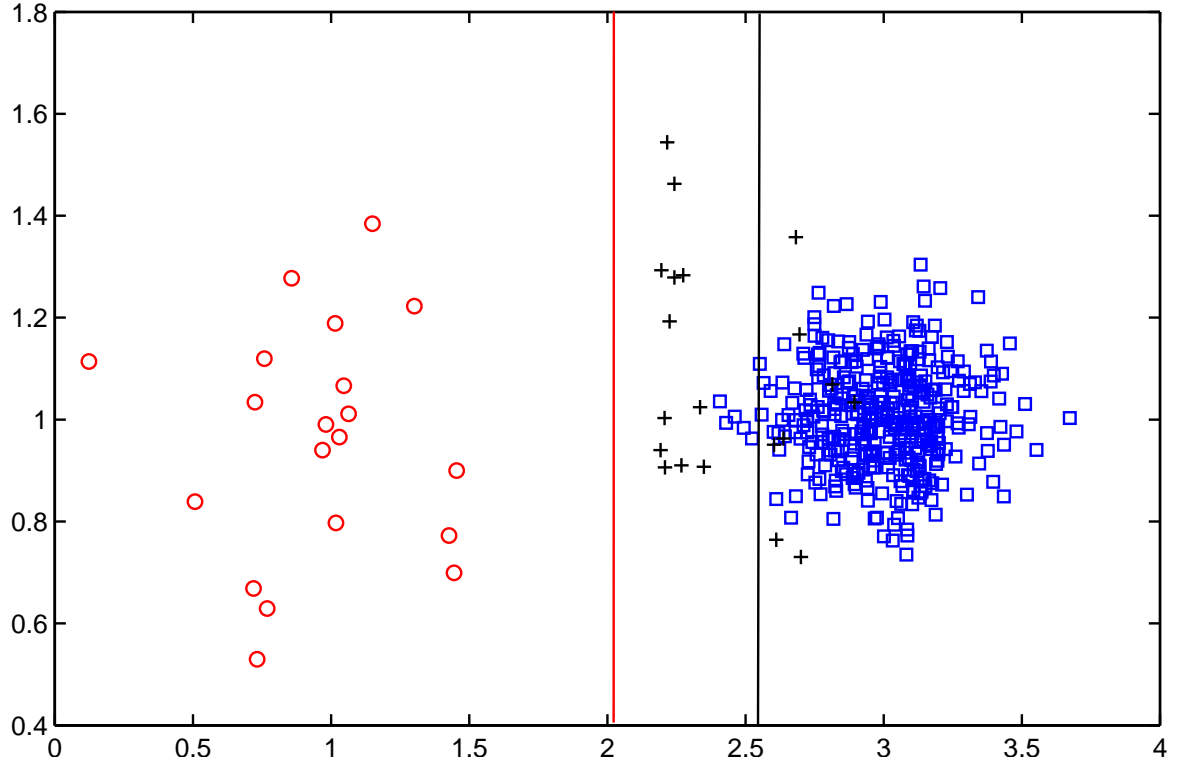
Figure 2.7:   Bootstrapping toy example.

less than a user-defined threshold. Performance saturation indicates that additional bootstrapping steps are unlikely to aid classifier performance further since the best-discriminating decision boundary within the saturation criteria has likely been found.

In this thesis, bootstrapping is used to help train a linSVM dismount detector (presented in Section 4.5.2).

2.6 Search Scheme Considerations for Spatial Detectors

Spatial detectors (i.e., detectors that explicitly or implicitly rely on spatial patterns of in-scene pixels to detect objects of interest) often require a search technique to determine which subset of image pixels should be evaluated. First, this section provides methodology for the simplest search technique: the sliding-window search scheme. Next, methodology for determining a measure of overlap between two windows (the coverage statistic) is provided. A technique for deconflicting alarm windows

that may be detecting the same object is provided next. Finally, general techniques for search-space reduction are provided.

    2.6.1 Sliding-window Search Scheme.   A common method for implementing a sliding-window search scheme is to generate a dense grid of overlapping windows at multiple scales [25]. A set of sliding-window parameters

$$\theta_w = \{w_x, w_y, h_{\min}, w_{\min}, \Delta s, \Delta x, \Delta y\},$$

is used to fully describe how the grid is to be implemented.

    The authors of [25] determined that the best set of sliding-window parameters to use for a HOG-based dismount detector is

$$\theta_w = \{w_x = 48, w_y = 96, h_{\min} = 72, w_{\min} = 0, \Delta s = 1.1, \Delta x = 0.1, \Delta y = 0.025\}.$$

Henceforth this thesis utilizes this set of parameter values when referring to θ_w.

    The base window, which will be scaled and shifted to produce the detection grid, is $w_x \times w_y$ pixels. The minimum height of a search window ($h_{\min}$) and the minimum width of a search window ($w_{\min}$) are used to compute the minimum scale value as follows:

$$s_{\min} = \max\left(\frac{h_{\min}}{w_y}, \frac{w_{\min}}{w_x}\right), \tag{2.10}$$

$$\therefore s_{\min} = \frac{h_{\min}}{w_y} \quad \because w_{\min} = 0 \text{ from } \theta_w. \tag{2.11}$$

    The maximum scale value is the largest scale value that will fit within the image boundaries ($x \in \mathbb{Z}[1, M]$ and $y \in \mathbb{Z}[1, N]$), determined as follows:

$$s_{\max} = \min\left(\frac{M}{w_x}, \frac{N}{w_y}\right). \tag{2.12}$$

For the purposes of this thesis, it is assumed that all images are wider than they are tall ($M \geq N$), which is commonly the case for imaging sensors. Since $w_x < w_y$ from θ_w, $\frac{M}{w_x}$ is guaranteed to be greater than $\frac{N}{w_y}$ if $M \geq N$. Therefore from Eqn. (2.12)

$$s_{\max} = \frac{N}{w_y}. \tag{2.13}$$

The scale ($\underline{s}$) is a geometric sequence with the *common ratio* (or multiplier) $\Delta s \in \mathbb{R}(1, \infty)$ with elements

$$s_n = s_{\min}(\Delta s)^{n-1}. \tag{2.14}$$

Since $s_n \leq s_{\max}$, the upper bound for n is derived from Eqns. (2.13)–(2.14) as

$$s_{\min}(\Delta s)^{n_{\max}-1} \leq s_{\max}, \tag{2.15}$$

$$(n_{\max} - 1)\ln(\Delta s) \leq \ln\left(\frac{s_{\max}}{s_{\min}}\right), \tag{2.16}$$

$$n_{\max} \leq \frac{\ln\left(\frac{s_{\max}}{s_{\min}}\right)}{\ln(\Delta s)} + 1, \tag{2.17}$$

$$\leq \left. \frac{\ln(s_{\max}) - \ln(s_{\min})}{\ln(\Delta s)} + 1 \right|_{s_{\max}=\frac{N}{w_y}, s_{\min}=\frac{h_{\min}}{w_y}},$$

$$\leq \frac{\ln(N) - \ln(w_y) - \ln(h_{\min}) + \ln(w_y)}{\ln(\Delta s)} + 1,$$

$$\leq \frac{\ln(N) - \ln(h_{\min})}{\ln(\Delta s)} + 1,$$

$$n_{\max} = \left\lfloor \frac{\ln(N) - \ln(h_{\min})}{\ln(\Delta s)} \right\rfloor + 1 \because n_{\max} \in \mathbb{Z}, \tag{2.18}$$

$$\therefore n \in \mathbb{Z}\left[1, \left\lfloor \frac{\ln(N) - \ln(h_{\min})}{\ln(\Delta s)} \right\rfloor + 1\right]. \tag{2.19}$$

For each scale (s_n), the search window is $s_n w_x \times s_n w_y$ pixels. The search window is then shifted through the x and y-directions using the shift multipliers (Δx and Δy)
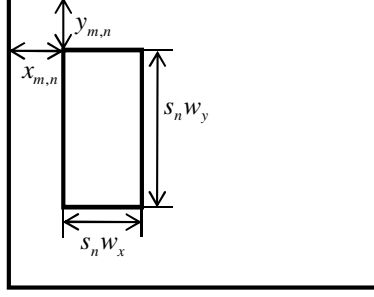
Figure 2.8:   Sliding window parameters example.

as follows:

$$x_{m,n} = 1 + (m-1)\Delta x s_n w_x, \forall x_m \le M - s_n w_x + 1, \qquad (2.20)$$

$$y_{m,n} = 1 + (m-1)\Delta y s_n w_y, \forall y_m \le N - s_n w_y + 1, \qquad (2.21)$$

where $x_{m,n}$ and $y_{m,n}$ are the top-left coordinates for the search window at scale s_n. Figure 2.8 depicts an example of how the parameters described in this section affect the size and location of the generated search windows.

2.6.2 Coverage Statistic.   It can be challenging to accurately determine the performance of a sliding-window detector for numerous reasons. First, ground-truth bounding boxes may be subjective based on the human that defines the bounding box limits. Furthermore, the size and location of the object in a ground-truth patch may not perfectly coincide with any detector window configuration.

For this reason, it is helpful to utilize a measure of overlap between two windows of arbitrary size and location. One such useful measure is the coverage statistic [25], defined as follows:

$$\Omega(a_i, a_j) = \frac{A(a_i \cap a_j)}{A(a_i \cup a_j)}, \qquad (2.22)$$

where a_i and a_j are rectangular windows of arbitrary size and location within the boundaries of the same image, $A(a_i \cap a_j)$ is the intersected area of the two windows,

2-19

$$\Omega(a_1, a_2) > \Omega(a_3, a_4)$$
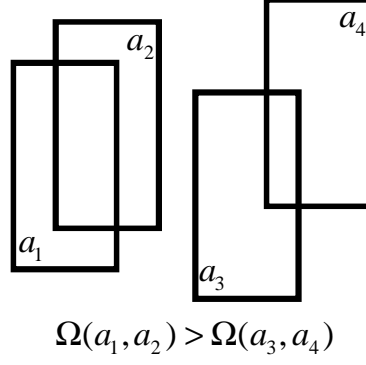
Figure 2.9:   Coverage statistic example.

and $A(a_i \cup a_j)$ is the union area of the two windows. If the windows have no overlap, then $A(a_i \cap a_j) = 0$ and consequently $\Omega(a_i, a_j) = 0$. If the two windows perfectly match, then $A(a_i \cap a_j) = A(a_i \cup a_j)$, and $\Omega(a_i, a_j) = 1$ as a result. Therefore, $\Omega(a_i, a_j) \in \mathbb{R}[0, 1]$. The coverage statistic concept is illustrated in Fig. 2.9.

2.6.3 Confidence-based Non-maximum Suppression of Detections.   Due to the nature of sliding-window detectors, it is possible that multiple search windows at similar location and size in the same image could result in multiple alarms for the same in-scene object. This can be problematic when trying to accurately gauge detector performance. In order to suppress the number of alarms produced by one object, it is useful to utilize a detection confidence output from the classifier for each alarm. For the purposes of linSVM, the magnitude of the classifier's real-valued output can be used as the confidence number.

First, it must be determined which alarms may be in conflict. For this, the coverage statistic is used [25]. If the coverage between two alarm windows from the same image (a_i and $a_j, i \neq j$) is greater than a threshold ($\eta_\Omega = 0.5$ as suggested by [25]), the windows are considered to be in conflict. For each conflict detected, the alarm window with the greater confidence is kept and the other is discarded. This process continues until all conflicts have been resolved. Figure 2.10 illustrates how multiple alarm windows (multiple colors on the left side of the figure) are suppressed

2-20
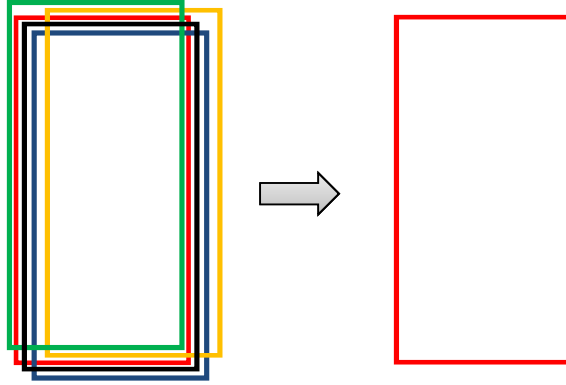
Figure 2.10: Confidence-based non-maximum suppression example. Multiple alarm windows that are considered to be in conflict (multiple colors on left) are suppressed leaving only the alarm window with the highest confidence value (red on right).

leaving only the alarm window with the highest confidence value (red on the right side of the figure).

It is possible that if two objects of interest are positioned very close to one another, two appropriate alarms may be considered to be in conflict by the confidence-based non-maximum suppression algorithm. In this case, an alarm may be falsely-suppressed. Figure 2.11 illustrates this scenario. The blue and red rectangles correspond to alarm windows for the green and pink dismounts respectively. The coverage of the two alarm windows is $\Omega = 0.5$, which is the exact threshold where two alarm windows are considered to be in conflict. In the worst-case scenario, both dismounts are viewed from a sagittal-plane (side-view) aspect. While neither dismount is partially-occluded (making them both valid targets for detector scoring), one of their respective alarm windows will likely be suppressed. This situation will result in a miss when scoring the detector.

2.6.4 Search Space Reduction Techniques for Sliding-window Detectors.
Many sliding-window detectors have a very large search area for each image under test. This often leads to significant computational costs which can limit the prospects of real-time processing [18], [19], [25], [46], [57], [61], [71].
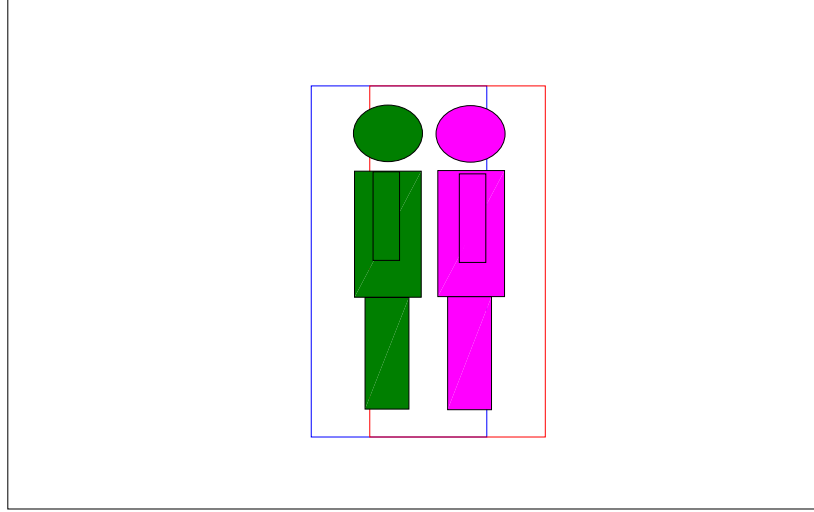
Figure 2.11:   Conflicting alarm window example. The blue rectangle represents an alarm window for the green dismount. The red rectangle represents an alarm window for the pink dismount.

The total size of the search space of an arbitrary $M \times N$ image (assume $M \geq N$ for this discussion) using sliding window parameter set θ_w can be derived from Eqns. (2.11)–(2.21). The total number of search windows (ς) per $M \times N$ image is derived as follows:

$$\varsigma = \sum_n \left\lfloor \frac{M - s_n w_x}{\Delta x s_n w_x} + 1 \right\rfloor \left\lfloor \frac{N - s_n w_y}{\Delta y s_n w_y} + 1 \right\rfloor. \tag{2.23}$$

For example, the parameter set θ_w (whose parameter values are defined in Section 2.6.1) results in $\varsigma \approx 1.85 \times 10^5$ search windows per 640×480 image. Intelligent reduction of this search space can dramatically improve overall processing speed, especially if the processing time for an individual search window is significant.

A very common method of search space reduction is to segment an image into foreground and background pixels. Foreground pixels are defined as pixels that should be identified using the detector of interest. Background pixels are defined as pixels that should be ignored by the detector.

2.6.4.1 Background Subtraction. A common method for segmenting foreground and background pixels is background subtraction. Numerous implementations of background subtraction are surveyed in [42]. Most modern background-subtraction techniques update the estimation of the background pixels over time.

Some background subtraction techniques make the assumption that the background does not change as a function of time and can therefore be determined a priori. The work in [42] indicates that such algorithms are of limited use in practical applications. It is logical that a priori background subtraction methods are at least limited to fixed observation platforms.

Overall, the advantages of background subtraction techniques are simplicity and speed. Notable disadvantages of time-dependent background subtraction algorithms is dependence on in-scene motion for detection of foreground pixels. Background subtraction systems that utilize a priori knowledge may be more capable of detecting stationary objects of interest if the a priori background model does not include those objects of interest.

A notable disadvantage of all background subtraction techniques is the problem of image registration between subsequent frames. This problem especially holds true for imaging systems mounted on mobile platforms or in high-vibration environments. Image registration requirements can negate the speed advantages of background subtraction algorithms.

2.6.4.2 Feature Cues. In-scene features can be used to determine foreground pixels in lieu of background subtraction [24], [27], [30], [37], [48], [65], [79]. One advantage of feature-based cues is that the features used can be custom-tailored for the class of object being detected. While the feature alone may not be sufficient to detect an object of interest–possibly due to false alarm sources or multiple instances of the same feature on one object–they may significantly reduce the search space for a more accurate spatial feature-based sliding-window detector.

Advantages of spectral feature cues may include speed (depending on the complexity of the feature being generated) and platform motion tolerance. Since spectral features are temporally independent, they can be used to define foreground pixels on each image frame independently. An additional advantage of temporal independence is that stationary objects of interest are relatively easy to find. In fact, stationary objects may be easier to find, depending on the sensor modality.

One potential disadvantage of spectral feature cues may include sensor modality issues. Sufficiently useful features may require exotic spectral bands or a large number of spectral bands. This may add significant cost to the system in terms of frame capture rate and/or monetary expense.

2.7 Skin Detection

This thesis effort proposes to utilize skin detection as a feature cue for reducing the search space of a HOG-based dismount detector. This section provides the background necessary about the spectral properties of skin and how they can be exploited for skin detection. First, a primer on reflectance and reflectance estimation is provided, followed by illumination considerations when developing spectral detection algorithms. Next, the spectral properties of human skin are provided, followed by features derived to take advantage of the spectral properties of skin for the purpose of skin detection and false alarm suppression. Methodology for extending these features to an arbitrary imager are provided next, followed by a basic skin detection algorithm based on the features described in this section.

2.7.1 Reflectance: Definition and Estimation.   Reflectance ($\rho_\lambda \in \mathbb{R}[0,1], \forall \lambda$) is defined as the percentage of incident electromagnetic power reflected by a material at wavelength λ. Many applications–especially in hyperspectral remote sensing–prefer to use imagery converted to the reflectance space since reflectance is an intrinsic material property that does not change based on illumination intensity or atmospheric
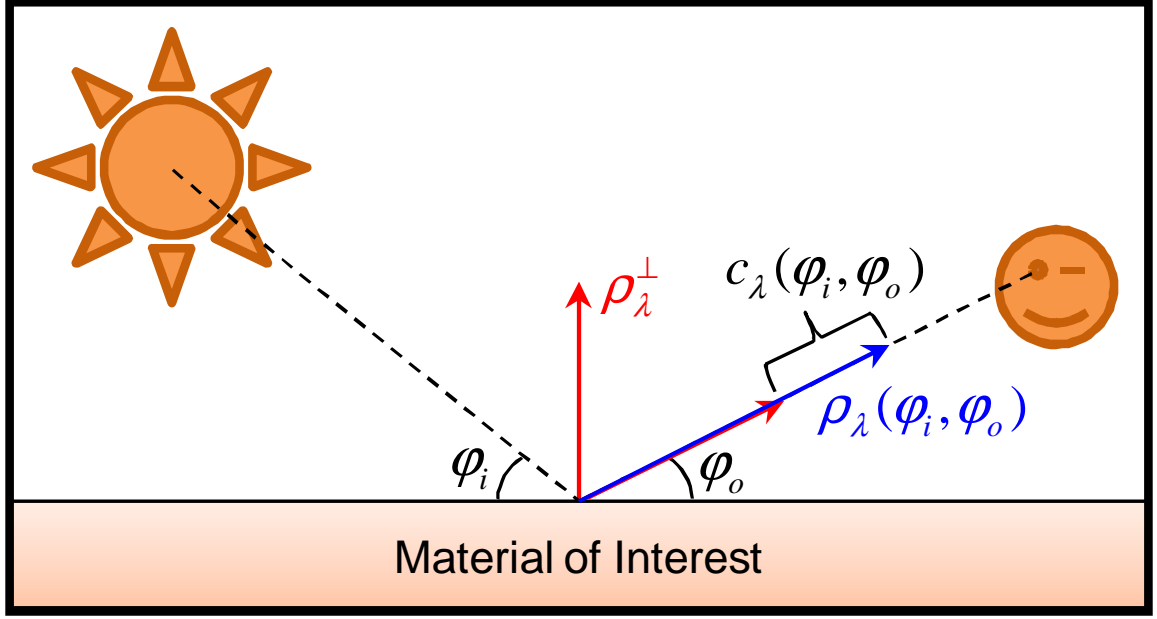
Figure 2.12:   Reflectance angular dependence.

variations[3]. However, reflectance *can* change as a function of illumination and observation angles [50]. For the purposes of this thesis, the full depth of this angular relationship is not explored. Instead, the angular relationship is incorporated as follows:

$$\rho_\lambda(\varphi_i, \varphi_o) = \rho_\lambda^\perp + c_\lambda(\varphi_i, \varphi_o), \tag{2.24}$$

where φ_i and φ_o are the incident and observation angles with respect to the material surface normal respectively (as depicted in Fig. 2.12); $\rho_\lambda^\perp$ is the material reflectance as measured by a reflectometer normal to the material surface; and $c_\lambda(\varphi_i, \varphi_o)$ is a function that encapsulates all angle dependence of the reflectance ($c_\lambda \in \mathbb{R}[-\rho_\lambda^\perp, 1 - \rho_\lambda^\perp]$). It should be noted that φ_i and φ_o can be further parameterized by their respective azimuth and elevation angles, thus making c_λ a four-dimensional function.

---

[3]Except in the case that illumination energy physically alters the material itself, whether from heating or induced chemical changes.

It is impossible to directly image a scene in reflectance space using passive sensors. Passive sensors require reflected or emitted energy from an in-scene object. Therefore, passive imagery is typically in radiance space, which is illumination dependent.

It is possible to transform an image from irradiance space to estimated reflectance space using one of several techniques. One method is to measure solar irradiance spectra at the time the image is acquired, then later divide the image irradiance spectra by the solar irradiance spectra. Another method is to estimate atmospheric absorption effects at the time of image acquisition, then cancel out those effects during post-processing.

One method for estimating atmospheric effects includes atmospheric modeling using a system such as MODTRAN [36] based on weather conditions recorded at the time and location of image acquisition. A simpler approach to correcting for atmospheric effects is to use a linear regression using an in-scene target of known reflectance (this method is also known as the empirical line method (ELM) [22]). The ELM for estimating reflectance ($\hat{\rho}_\lambda$) is implemented as

$$\hat{a}_\lambda = \frac{\mu_\lambda^w - \mu_\lambda^b}{\rho_\lambda^w - \rho_\lambda^b}, \tag{2.25}$$

$$\hat{b}_\lambda = \frac{\mu_\lambda^b \rho_\lambda^w - \mu_\lambda^w \rho_\lambda^b}{\rho_\lambda^w - \rho_\lambda^b}, \tag{2.26}$$

$$\hat{\rho}_\lambda = \frac{X_\lambda - \hat{b}_\lambda}{\hat{a}_\lambda}, \tag{2.27}$$

where X_λ is the input image at wavelength λ in intensity space; ρ_λ^w and ρ_λ^b are the known reflectances of a bright and dark in-scene object respectively; and μ_λ^w and μ_λ^b are the average image intensity values of the same bright and dark in-scene objects respectively. If only one object of known reflectance is available, Eqn. (2.27) can be simplified to

$$\hat{\rho}_\lambda = \frac{\rho_\lambda^w}{\mu_\lambda^w} X_\lambda, \tag{2.28}$$

which assumes that an image intensity value of 0 corresponds to a reflectance value of 0. This assumption does not necessarily hold true (especially when sensor noise is considered), but is often accurate enough to be useful.

There are a few key issues that must be considered when using the technique described in Eqn. (2.27) or (2.28). First, the ELM method assumes that all image pixels are receiving identical illumination. There are many obvious situations where this assumption is false, but in practice, ELM is still very effective for estimating reflectance when it is possible to operate in conditions as close to this assumption as possible.

The ELM method also assumes that the relationship between image intensity and reflectance is linear. Depending on the sensor being used, this may or may not be a valid assumption to make. While the linear relationship may not necessarily hold true, non-linearity in the relationship tends to be minor and of little consequence in practice.

Another key issue to consider is image saturation. In bright illumination conditions, it is possible for image intensity values for some pixels to be saturated at the maximum allowable value. This saturation condition affects the accuracy of ELM estimation because there is no way to know what the true values of saturated pixels are. This is especially a problem if the saturated pixels are on the calibration object itself, which can drastically affect the reflectance estimation of every pixel in the image.

To mitigate the saturation issue, the imaging sensor should be operated such that no pixels are saturated. In the event that the operator cannot control image gain or other parameters that may mitigate the saturation issue (e.g. the sensor uses "auto-gaining" to set the brightest pixel to the maximum value), a saturation target should be placed in the scene being imaged. The saturation target should be significantly brighter than the bright ELM calibration object and placed such that the saturation target does not "wash out" areas of interest in the scene or cause secondary illumination of the calibration targets or areas of interest in the scene.
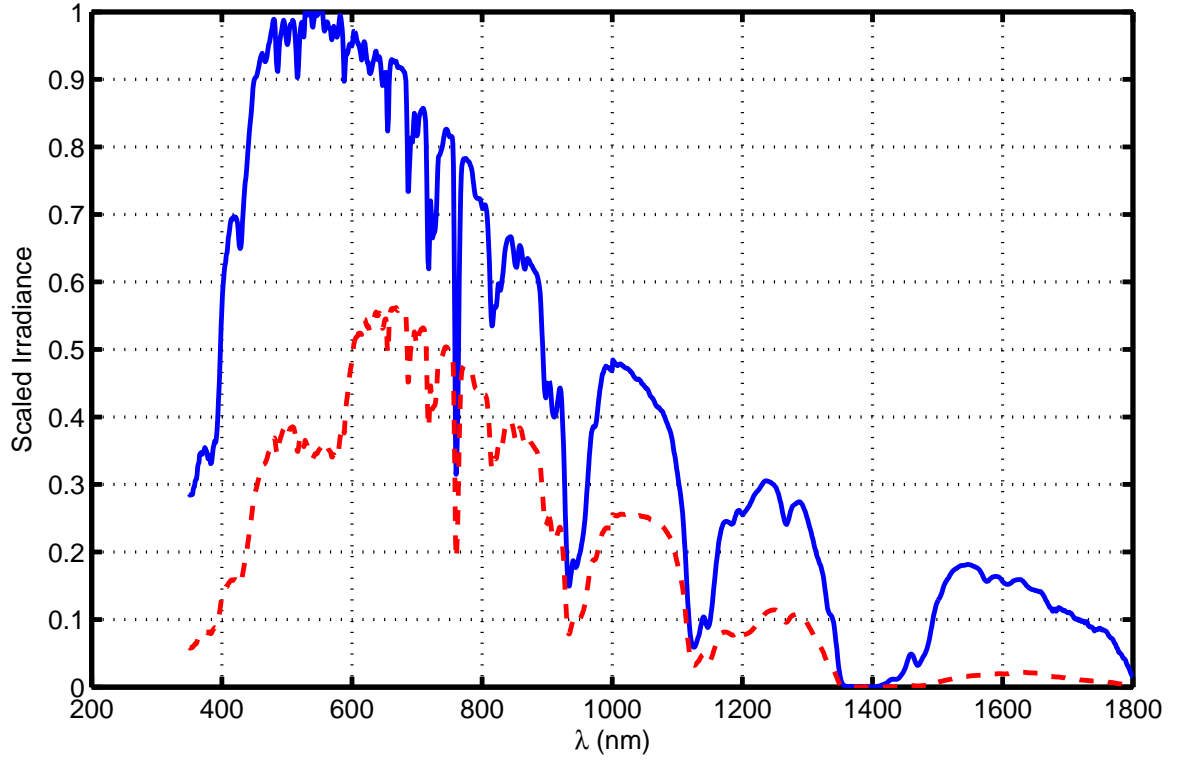
Figure 2.13:   Solar irradiance in Dayton, OH on a sunny day scaled by the maximum
irradiance (solid blue) and the irradiance spectra of light-complected
skin illuminated by sunlight scaled by the same maximum irradiance
(dashed red) from [51], [53].

2.7.2   Illumination Considerations.   When developing features for hyperspec-
tral detection applications, it is important to consider limitations of the illumination
source. Solar illumination is often used when remotely estimating reflectance val-
ues from a hyperspectral camera. Figure 2.13 depicts measured solar irradiance in
Dayton, OH on a sunny day scaled by the maximum irradiance (solid line). The
dashed line in Fig. 2.13 is the measured irradiance spectra of light-complected skin
illuminated by sunlight scaled by the same maximum irradiance.

Note that there are areas of extremely-low solar irradiance at the earth's surface
near 1400nm and 1900nm. These troughs correspond to atmospheric water absorp-
tion. Since solar illumination is very poor at these wavelengths, they should be avoided
for use in any solar-illuminated detection algorithm.

2-28
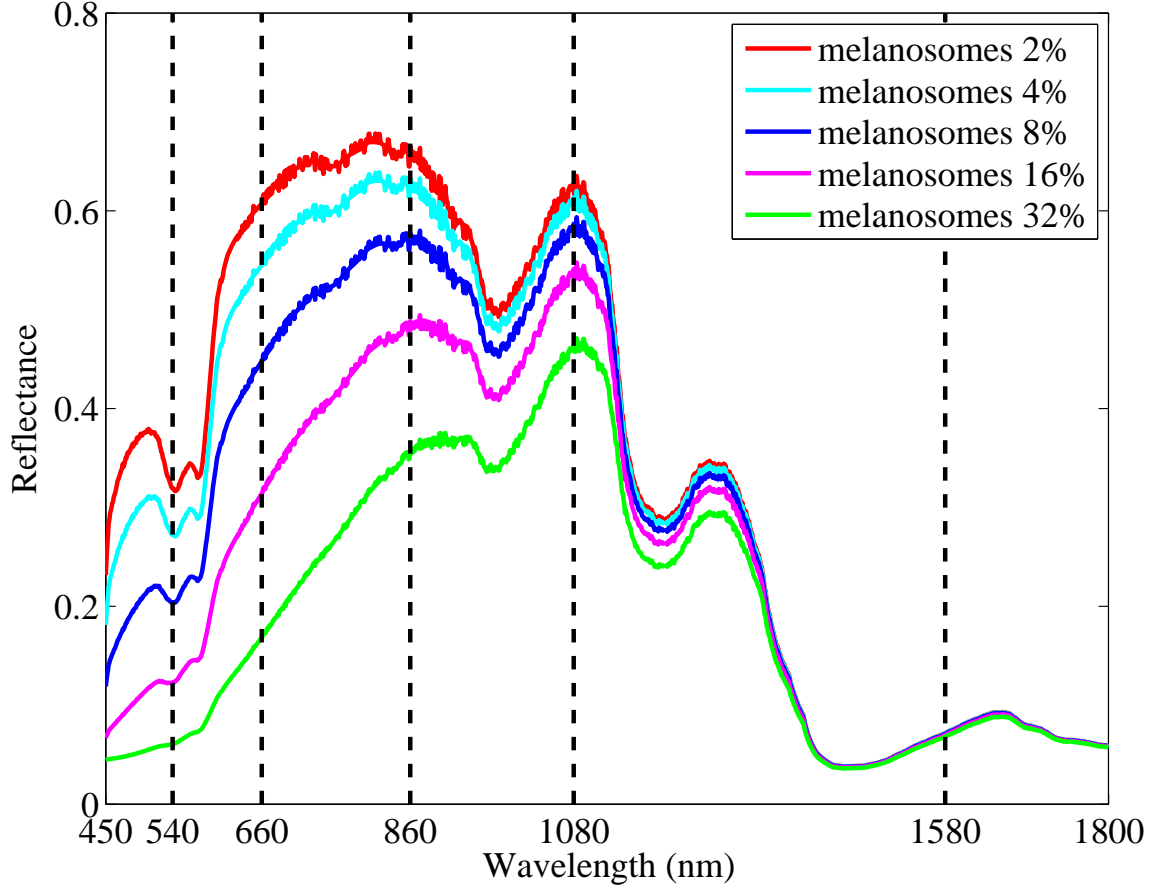
Figure 2.14:   Model-generated skin reflectance spectra from [51], [55].

2.7.3 Properties of Human Skin.   Human skin exhibits numerous distinctive absorption features in the visible (VIS) and NIR regions of reflectance spectra. These absorption features can be exploited for detecting skin [51], [52], [53]. Figure 2.14 depicts several examples of modeled skin reflectance for various levels of skin pigmentation (including the extremes), and the relevant wavelengths used for skin detection algorithms discussed in Section 2.7.8.

It is important to note that there is a distinct drop-off in skin reflectance beyond 1150nm, with local maxima at 1080nm and 1250nm; and local minima at 1200nm and 1400nm. These features are primarily due to water absorption [9]. Based on these skin-reflectance observations, useful descriptive features can be generated using the

following:

$$Q = \frac{\rho_{\lambda_1} - \rho_{\lambda_2}}{\rho_{\lambda_1} + \rho_{\lambda_2}}, \lambda_1 \neq \lambda_2, \tag{2.29}$$

where the feature (Q) is a difference of reflectance at wavelengths λ_1 and λ_2 normalized by the sum of the reflectance at those respective wavelengths. Since $\rho_\lambda \in \mathbb{R}[0, 1], \forall \lambda$, the numerator of Eqn. (2.29) must be $\in \mathbb{R}[-1, 1]$. Since the magnitude of the denominator of Eqn. (2.29) will always be greater than or equal to the magnitude of the numerator (except in the statistically improbable case that $\rho_{\lambda_1} = \rho_{\lambda_2} = 0$), the absolute value of Q must be less than or equal to 1. Therefore

$$Q \in \mathbb{R}[-1, 1] \iff (\rho_{\lambda_1} > 0) \vee (\rho_{\lambda_2} > 0). \tag{2.30}$$

Equation (2.29) is a generalization inspired by the normalized difference vegetation index (NDVI) [23], which is used for remote detection of vegetation.

2.7.4 Normalized Difference Skin Index (NDSI). The large drop-off in skin reflectance between 1080nm and 1400nm is an excellent candidate for generating a useful feature from Eqn. (2.29). Additionally, the relative stability of skin reflectance values at these wavelengths across the gamut of human skin types (as evidenced in Fig. 2.14) is also useful. However, as noted in Section 2.7.2, reflectance near 1400nm should be avoided for generating features for detection purposes. Therefore, the next available wavelength above 1400nm that has sufficient solar irradiance–1580nm–is used [51], [53].

The normalized difference skin index (NDSI) [51], [53] value (γ) is derived from Eqn. (2.29) as

$$\gamma = \frac{\rho_{\lambda_1 = 1080\text{nm}} - \rho_{\lambda_2 = 1580\text{nm}}}{\rho_{\lambda_1 = 1080\text{nm}} + \rho_{\lambda_2 = 1580\text{nm}}}. \tag{2.31}$$

2-30
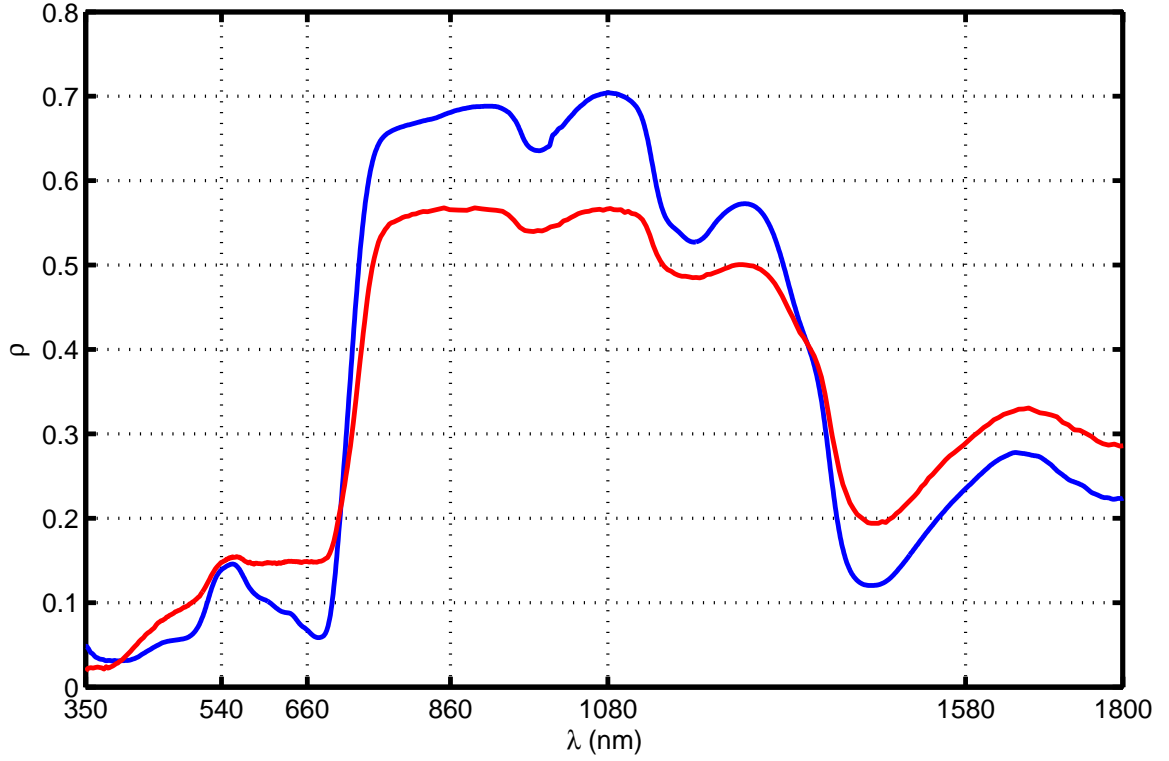
Figure 2.15:   Reflectance spectra of lodgepole pine (blue) and dry grass (red) from [15].

It is possible that other materials with water-absorption features similar to skin may be false alarm sources for a detector based solely on the NDSI. Such materials include certain kinds of vegetation (especially in the yew family) as illustrated in Fig. 2.15; and materials with high water content and back-scattering properties (e.g., snow) as illustrated in Fig. 2.16.

2.7.5   Normalized Difference Vegetation Index (NDVI).   A commonly-used feature for detecting vegetation is the NDVI [23], defined as

$$\alpha = \frac{\rho_{\lambda_1 = 860\mathrm{nm}} - \rho_{\lambda_2 = 660\mathrm{nm}}}{\rho_{\lambda_1 = 860\mathrm{nm}} + \rho_{\lambda_2 = 660\mathrm{nm}}}, \tag{2.32}$$

where α is the NDVI value.

The NDVI feature takes advantage of a typically large derivative in vegetation reflectance spectra between approximately 660nm and 860nm, as can be seen in

2-31
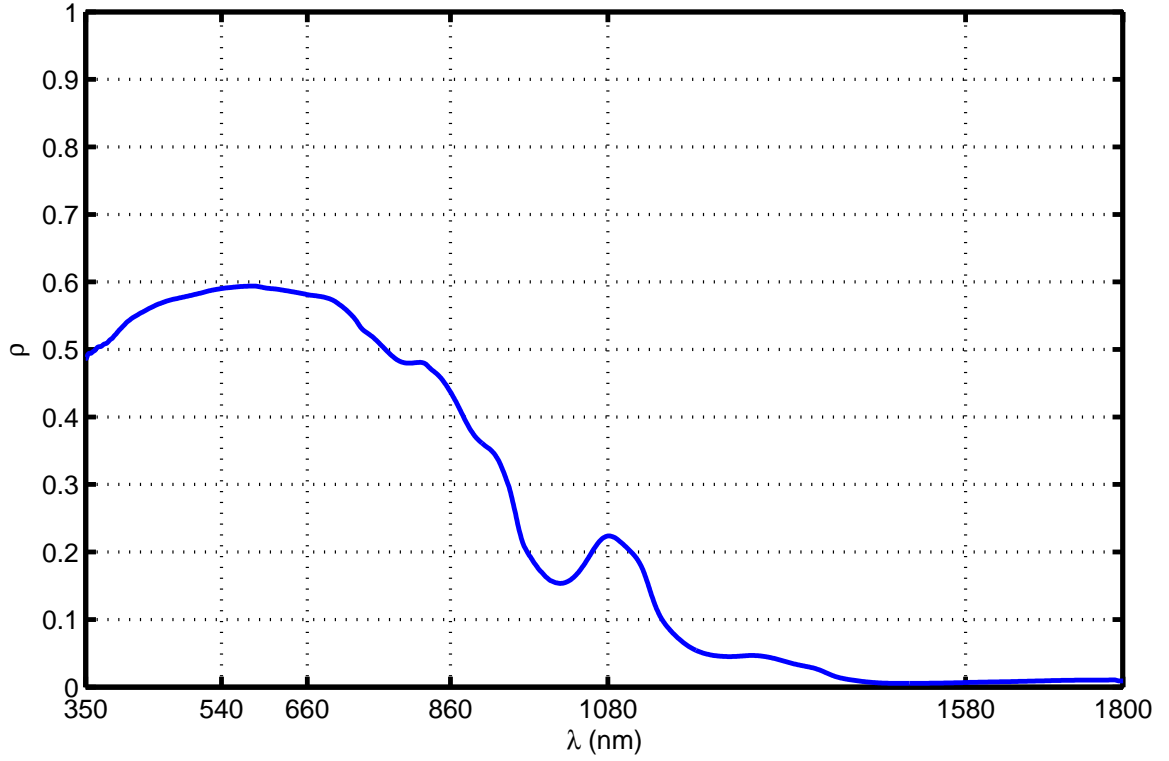
Figure 2.16:   Reflectance spectra of snow from [15].

Fig. 2.15. The lower reflectance values near 660nm are due to chlorophyll absorption, while the higher reflectance values near 860nm are a result of high scattering in the NIR. It is possible that the NDVI may be useful for suppressing false alarms produced by an NDSI-based skin detector.

2.7.6   Normalized Difference Green Red Index (NDGRI).   It is observed in Fig. 2.14 that healthy human skin is more red than green. It is observed in Fig. 2.15 that healthy vegetation (blue curve) is more green than red and dry vegetation (red curve) is close to equal for the red and green components. The drop in red reflectance for healthy vegetation is due to chlorophyll absorption. It is observed from Fig. 2.16 that the red and green components of snow are relatively equal. This knowledge of green-red ratio can be useful for generating another feature for suppressing false alarms produced by an NDSI-based skin detector.

Since the green and red components of many RGB cameras are nominally centered at 540nm and 660nm respectively, the normalized difference green-red index (NDGRI) feature (β) can be derived from Eqn. (2.29) as

$$\beta = \frac{\rho_{\lambda_1=540\text{nm}} - \rho_{\lambda_2=660\text{nm}}}{\rho_{\lambda_1=540\text{nm}} + \rho_{\lambda_2=660\text{nm}}}. \tag{2.33}$$

2.7.7 Extending Features to an Arbitrary Imaging System. The skin features described previously depend on having perfect knowledge of the reflectance of human skin. In the case of an imaging scenario, many factors affect the estimation of reflectance spectra. These include, but are not limited to, uncertainty in atmospheric correction [77], sensor noise, and specular reflection (as noted in Section 2.7.1).

Evidence from [41] indicates that skin is a lambertian surface ($\rho_\lambda = \rho_\lambda^\perp$) if the illumination source is perpendicular to the tissue surface. This same article shows that skin is highly forward-scattering ($c_\lambda(\varphi_i, \varphi_o) > 0$) as the illumination angle decreases from perpendicular to the surface of the skin (as depicted in Fig.2.12).

A typical signal-plus-noise model is derived to approximate how sensor noise and specular reflection affect reflectance values and consequently generated-feature values. The signal-plus-noise model for estimated reflectance is

$$\hat{\rho}_\lambda = \rho_\lambda^\perp + c_\lambda + n_\lambda \tag{2.34}$$

where $\hat{\rho}_\lambda$ is the estimated reflectance from the imager, c_λ is the specular reflection term, and n_λ is an assumed-Gaussian noise term distributed as $\mathcal{N}(0, \sigma_\lambda^2)$ (note that the noise term is modeled in the reflectance space).

Consider the effects the specular and noise components have on Eqn. (2.29):

$$Q = \frac{\left(\rho_{\lambda_1}^\perp + c_{\lambda_1} + n_{\lambda_1}\right) - \left(\rho_{\lambda_2}^\perp + c_{\lambda_2} + n_{\lambda_2}\right)}{\left(\rho_{\lambda_1}^\perp + c_{\lambda_1} + n_{\lambda_1}\right) + \left(\rho_{\lambda_2}^\perp + c_{\lambda_2} + n_{\lambda_2}\right)}. \tag{2.35}$$

For the sake of discussion, it is assumed that the specular reflection is independent of wavelength. That is, $c_\lambda = c, \forall \lambda$, and for all pixels in the image. It is further assumed that the distribution of the noise *is* wavelength-dependent. Given these assumptions, Eqn. (2.35) is simplified as:

$$Q = \frac{\left(\rho_{\lambda_1}^\perp - \rho_{\lambda_2}^\perp\right) + (n_{\lambda_1} - n_{\lambda_2})}{\left(\rho_{\lambda_1}^\perp + \rho_{\lambda_2}^\perp\right) + (n_{\lambda_1} + n_{\lambda_2}) + 2c}. \tag{2.36}$$

Under the assumption that each noise term is drawn from a zero-mean normal distribution, $E[\mathcal{N}(0, \sigma_\lambda^2)] = 0$, and,

$$E[Q] = \frac{\left(\rho_{\lambda_1}^\perp - \rho_{\lambda_2}^\perp\right)}{\left(\rho_{\lambda_1}^\perp + \rho_{\lambda_2}^\perp\right) + 2c}. \tag{2.37}$$

As can be seen from Eqn. (2.37), a significant amount of specular reflection in a pixel can significantly lower that pixel's NDSI, NDVI, and NDGRI values. Furthermore, even though the expected value of the noise term is zero, sensor noise will still affect the normalized difference terms as suggested in Eqn. (2.36). The larger the noise power, the greater impact seen in the imaged data.

2.7.8 Rules-based Skin Detection Algorithms.   The rules-based detector utilizes the NDSI values for skin detection and either NDVI or NDGRI values to suppress detections of potential skin confusers such as vegetation and snow [51], [52], [55].

A rules-based skin detector based on NDSI and NDVI is defined as

$$S_i : i = \begin{cases} 1 & \text{if } \alpha \in \mathbb{R}[a_1, a_2] \text{ and } \gamma \in \mathbb{R}[c_1, c_2] \\ 0 & \text{otherwise} \end{cases}, \tag{2.38}$$

where $a_1, a_2, c_1,$ and c_2 are the limits of a rectangular decision region in two-dimensional (α, γ) space.

Similarly, a rules-based skin detection based on NDSI and NDGRI is defined as:

$$S_i : i = \begin{cases} 1 & \text{if } \beta \in \mathbb{R}[b_1, b_2] \text{ and } \gamma \in \mathbb{R}[c_1, c_2] \\ 0 & \text{otherwise} \end{cases} . \tag{2.39}$$

where b_1, b_2, c_1, and c_2 are the limits of a rectangular decision region in two-dimensional (β, γ) space.

The advantage the simple detectors described here is the dependence solely on the extremes in skin spectral measurements. Given the availability of the model in [55], these spectra are generated with a high degree of confidence. The upper and lower bounds on the values for α, β, and γ computed using the skin model and are: $a_1 = -0.003891$, $a_2 = 0.50321$, $b_1 = -0.54079$, $b_2 = -0.061525$, $c_1 = 0.65703$, and $c_2 = 0.76779$. It further has the advantage that one only the target information must be taken into account. In the case of the detector described in Eqn. (2.38) and Eqn. (2.39), the decision region is rectangular. In order to generate a ROC curve, (α, γ) for (NDVI,NDSI) and (β, γ) for (NDGRI,NDSI) must be varied across their respective ranges yielding a two-dimensional ROC curve surface (or choose a few operating points and determine several one-dimensional ROC curves). Two primary limitations of this approach are that it does not take into account information on potential false alarm sources beyond the design of the normalized difference indices, and it ignores the distribution of the target and false alarm sources, therefore lacking optimality in terms of minimizing the Bayes risk.

2.8   Classic Detection Theory

This section provides background on classic detection theory and a method for estimating the probability density function (pdf) of a set of incomplete data.

2.8.1   Likelihood Ratio Test.   Binary detectors are often used to determine if a random sample belongs to the positive or negative class. If the pdf of all samples in the positive class ($f_1(x)$, where $x \in X$ and X is the distribution of observations)

is known and the pdf of all samples in the negative class ($f_0(x)$) is known, a simple LRT can be devised to hypothesize whether a randomly-observed sample is within the positive or negative class [17]. Recall from Section 2.1 that $\mathcal{S}$ is the space that contains all possible observations. Therefore, $X \in \mathcal{S}$.

The hypothesis that a sample lies within the positive class is defined as H_1, while the hypothesis that a sample lies within the negative class is defined as H_0. Cost factors ($C_{ij}; i, j \in \{0, 1\}$) represent the relative costs of declaring that H_i is true given that H_j is actually true.

With the above definitions, it is now possible to define the Bayes risk

$$\mathcal{R} \equiv E[\text{cost}] = \sum_{i,j=\{0,1\}} C_{ij} P[H_i|H_j] P_j, \tag{2.40}$$

where $P[H_i|H_j]$ is the probability of declaring that H_i is true given that H_j is true, P_1 is the prior probability that any arbitrary sample will be in the positive class, and P_0 is the prior probability that any arbitrary sample will be in the negative class ($P_1 \in \mathbb{R}[0, 1]$, $P_0 \in \mathbb{R}[0, 1]$, and $P_1 + P_0 = 1$).

For the decision regions

$$S_i : i = \begin{cases} 1 & \text{if evidence suggests that } H_1 \text{ is true} \\ 0 & \text{if evidence suggests that } H_0 \text{ is true} \end{cases}, \tag{2.41}$$

it must be determined how the choice of decision rule affects the Bayes risk. It is now possible to define $P[H_i|H_j]$ in terms of the decision regions: $P[H_i|H_j] = \int_{S_i} f_j(x) dx$. Substituting this definition into Eqn. (2.40) yields

$$\mathcal{R} = \sum_{i,j=\{0,1\}} C_{ij} P_j \int_{S_i} f_j(x) dx. \tag{2.42}$$

Note that for an arbitrary pdf ($f(x)$), the following holds true:

$$\int_{S_1} f(x)dx + \int_{S_0} f(x)dx = \int_{\mathcal{S}} f(x)dx = 1, \tag{2.43}$$

$$\therefore \int_{S_1} f(x)dx = 1 - \int_{S_0} f(x)dx. \tag{2.44}$$

From the generality determined in Eqn. (2.44), Eqn. (2.42) can be rewritten and expanded in terms of only S_0 as

$$\mathcal{R} = C_{10}P_0 + C_{11}P_1 + \int_{S_0} \underbrace{[(C_{00}P_0 - C_{10}P_0) f_0(x) - (C_{11}P_1 - C_{01}P_1) f_1(x)]}_{\dot{\mathcal{R}}} dx.$$

$$\tag{2.45}$$

For an optimal decision rule, the Bayes risk should be minimized. To minimize Eqn. (2.45), any $x \in \mathcal{S}$ that results in a negative value for $\dot{\mathcal{R}}$ should be included in the decision region S_0. Therefore,

$$x \in S_0 \iff (C_{00}P_0 - C_{10}P_0) f_0(x) - (C_{11}P_1 - C_{01}P_1) f_1(x) < 0, \tag{2.46}$$

$$\iff (C_{00}P_0 - C_{10}P_0) f_0(x) < (C_{11}P_1 - C_{01}P_1) f_1(x), \tag{2.47}$$

$$\iff \frac{(C_{00} - C_{10})P_0}{(C_{11} - C_{01})P_1} > \frac{f_1(x)}{f_0(x)}. \tag{2.48}$$

Since $x \in S_1 \iff x \notin S_0$, the reverse of the inequality in Eqn. (2.48) holds true for $x \in S_1$.

The *likelihood ratio* is defined as

$$\Lambda_X(x) \equiv \frac{f_1(x)}{f_0(x)}. \tag{2.49}$$

The LRT threshold is defined as

$$\eta_\Lambda \equiv \frac{(C_{00} - C_{10})P_0}{(C_{11} - C_{01})P_1}. \tag{2.50}$$

Combining Eqns. (2.48)–(2.50), the LRT decision rule is

$$S_i : i = \begin{cases} 1 & \text{if } \Lambda_X(x) > \eta_\Lambda \\ 0 & \text{if } \Lambda_X(x) < \eta_\Lambda \end{cases}.$$

(2.51)

Since $\Lambda_X(x) = \eta_\Lambda$ partitions $\mathcal{S}$ into S_1 and S_0, it is known as the *decision boundary.*

2.8.2 Expectation Maximization for Gaussian Mixture Models. Accurately estimating the pdf of a random data set is a difficult problem [21], especially if an incomplete set of observations is available. Even representing the functional form of the pdf can be daunting depending on the distribution of the data set. A *mixture model* is a weighted combination of multiple simple pdfs for the purpose of approximating any arbitrarily complex pdf.

One of the most commonly-used mixture models is the Gaussian mixture model (GMM), which is a weighted combination of multiple Gaussian pdfs. The advantage of the GMM is that a Gaussian pdf can be efficiently described using a relatively compact set of sufficient statistics (namely the mean (μ) and the variance (σ^2)). Figure 2.17 depicts a toy example. The red dashed curves are two arbitrary Gaussian pdfs (the left curve is $\mathcal{N}(-2, 1)$ and the right curve is $\mathcal{N}(1, 9)$). The solid blue curve is the weighted sum of the red dashed curves, where the left curve has a weight of 0.2 and the right curve has a weight of 0.8.

Estimating the parameters for a GMM based on an incomplete set of observations from a data set can be accomplished in a number of ways. The expectation maximization (EM) algorithm [47] is a useful (albeit suboptimal) method for estimating GMM parameters. The EM-GMM algorithm is a two-stage iterative process as outlined in Fig. 2.18.

First, there is an initialization step where an initial guess of the GMM parameters is provided. This includes the number of Gaussians used in the GMM (K), the means of each Gaussian ($\mu_k, k = 1, ..., K$), the variances of each Gaussian (σ_k^2), and
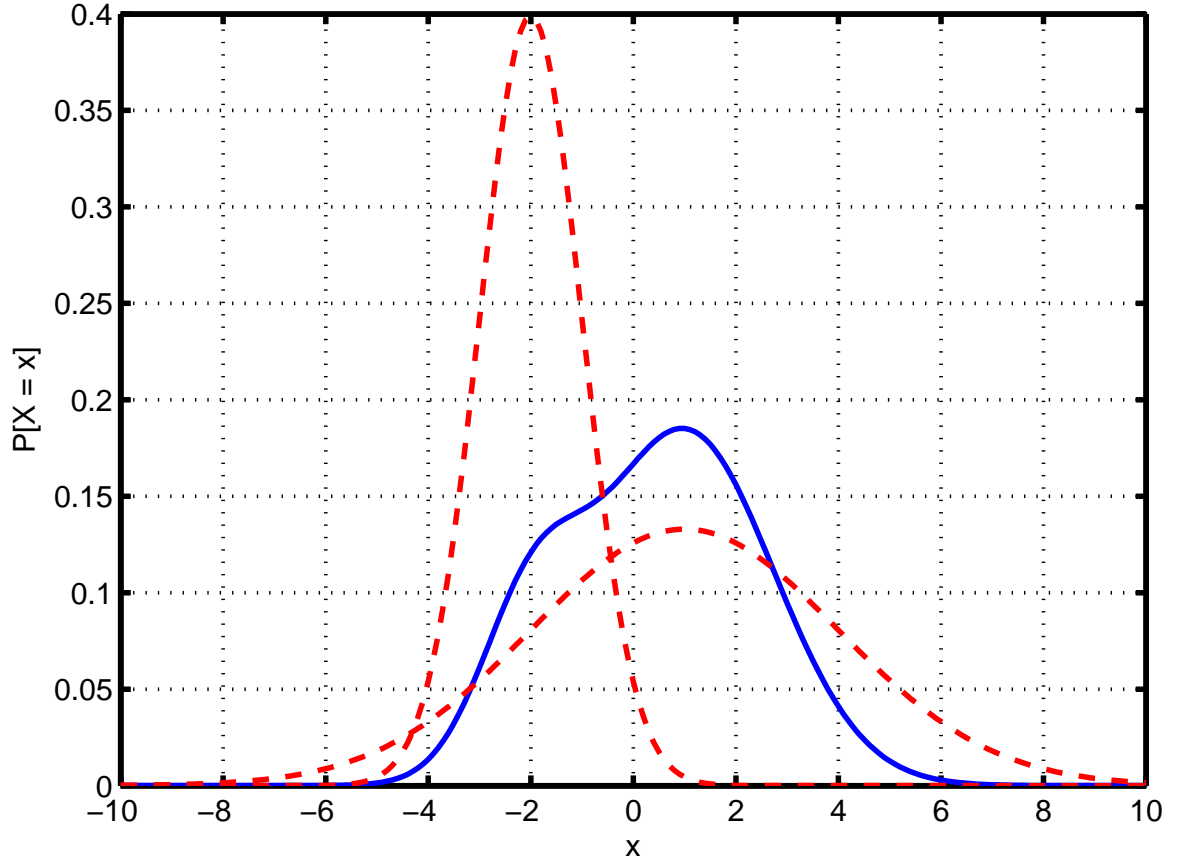
Figure 2.17: Gaussian mixture model toy example. The red dashed curves are two arbitrary Gaussian pdfs. The solid blue curve is the weighted sum of the red dashed curves, where the left curve ($\mathcal{N}(-2, 1)$) has a weight of 0.2 and the right curve ($\mathcal{N}(1, 9)$) has a weight of 0.8.
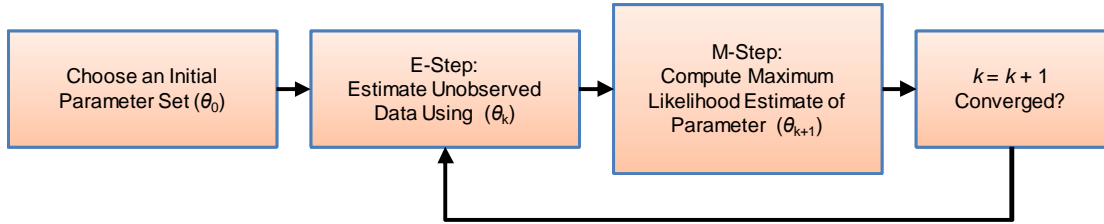
Figure 2.18: Expectation maximization (EM)- Gaussian mixture model (GMM) flowchart inspired by [47].

the weights of each Gaussian ($\pi_k \in \mathbb{R}[0, 1], \sum_{k=1}^{K} \pi_k = 1$). Note that the initial guess for K is not updated by the algorithm whereas the other parameters are.

In the expectation stage, the probability that the kth Gaussian occurred given the set of observations $(x_m, m = 1, ..., M)$ is evaluated as

$$\Phi_{(m,k)} = \frac{\pi_k f_k(x_m)}{\sum_{k=1}^{K} \pi_k f_k(x_m)}, \tag{2.52}$$

where $f_k(x_m)$ is the functional evaluation of the kth Gaussian pdf at the point x_m.

After the expectation stage is complete, all Gaussian parameters and weights are re-estimated in the maximization stage by

$$M_k = \sum_{m=1}^{M} \Phi_{(m,k)}, \tag{2.53}$$

$$\pi_k = \frac{M_k}{M}, \tag{2.54}$$

$$\mu_k = \frac{\sum_{m=1}^{M} x_m \Phi_{(m,k)}}{M_k}, \tag{2.55}$$

$$\sigma_k^2 = \frac{\sum_{m=1}^{M} \Phi_{(m,k)}(x_m - \mu_k)^2}{M_k}, \tag{2.56}$$

where M_k is a temporary normalization term.

After the maximization stage is complete, the GMM parameters are checked against a convergence criterion. Typically, convergence is measured by considering the log-likelihood of the current parameter set $(\underline{\theta} = \{\underline{\pi}, \underline{\mu}, \underline{\sigma}^2\})$ or the likelihood that the GMM with the current parameter set produced the measured data as

$$\ell(\underline{\theta}|\underline{x}) = \ln\left(\prod_{m=1}^{M} \sum_{k=1}^{K} \pi_k f_k(x_m)\right),$$

$$= \sum_{m=1}^{M} \ln\left(\sum_{k=1}^{K} \pi_k f_k(x_m)\right). \tag{2.57}$$

If $\underline{\theta}$ or $\ell(\underline{\theta}|\underline{x})$ has met their respective convergence criterion, the process stops. Otherwise, the algorithm loops back to the expectation stage. Typical convergence cri-

teria include: $\underline{\theta}$ or $\ell(\underline{\theta})$ does not deviate beyond some ϵ (i.e., *stationarity* is achieved), or some predefined number of training steps has occurred.

Some advantages to note about the EM-GMM algorithm are that it is simple, it is stable, there are no learning parameters (as used with gradient descent), Hessians are not required, likelihood increases at each iteration, and the maximum likelihood value cannot be "overshot". One disadvantage is that only a local maximum for the likelihood can be obtained, thus the algorithm is not guaranteed to return an optimal solution. Additionally, the local maximum that is found is sensitive to the initialization of the parameters. Finally, the algorithm is computationally expensive.

In this thesis, the EM-GMM algorithm is used to approximate the distribution of skin and non-skin samples in feature space (presented in Section 3.2.3).

2.9   Summary

This chapter presents the background information necessary for this thesis. The chapter begins with a description of the notation used throughout this thesis. Next, the basic dismount tracking architecture is presented, followed by descriptions of passive sensors commonly used for tracking purposes.

State-of-the-art dismount detection techniques are presented next, including an in-depth discussion of HOG features, linSVM, and bootstrapping. This discussion is followed by search space considerations for sliding window detectors, which leads to the defining purpose of this thesis: utilizing skin detection to reduce the search space of a sliding window detector.

Next, the properties of human skin and how those properties are exploited for robust skin detection are presented. Finally, background on classic detection theory is presented since it is necessary for a skin detection technique described in Chapter III.
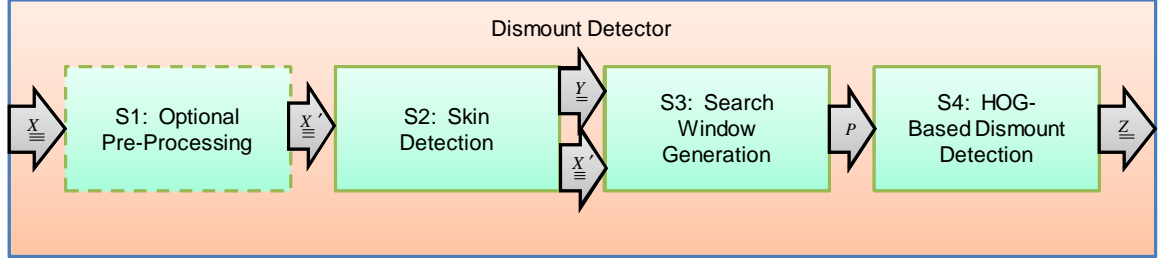
Figure 3.1: Block diagram of the proposed skin-detection-cued dismount detection system.

III. Methodology

As mentioned in Chapter I, the primary focus of this thesis is dismount detection. The proposed dismount detection system presented in this chapter employs recent efforts in human skin detection to cue a robust, spatial-feature-based dismount detector. The goal is to reduce the search space required for the spatial-feature-based dismount detector while suppressing potential false alarm sources.

The block diagram depicted in Fig. 3.1 provides an overview of the proposed process. The proposed dismount-detection system occurs in four stages. The first stage is an optional pre-processing stage for input imagery. The second stage is detection of skin pixels in an image. The third stage generates search windows within the image based on the locations of detected skin. The fourth stage runs a histograms of oriented gradients (HOG)-based dismount detector on each search window generated in the third stage.

3.1 S1: Optional Pre-processing Stage

In the optional pre-processing stage depicted in Fig. 3.2, any sensor-specific image pre-processing occurs. For example, it may be necessary to process the imagery to account for aberrations induced by the sensor. These aberrations can include, but are not limited to, non-uniformity, bad pixels, and sensor noise.

In particular, it may be necessary to incorporate power thresholding to account for noise. In image pixels where signal power is very near or below the sensor noise
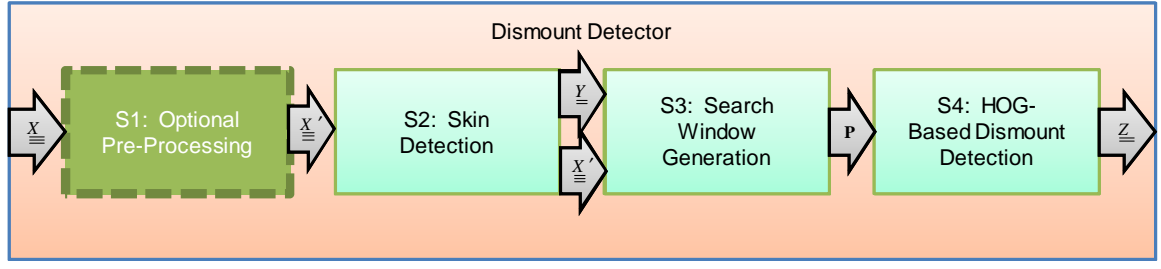
Figure 3.2: Stage 1: Optional pre-processing.

floor (deep shadows for example), the noise component of the pixel value may dominate subsequent calculations (this is a known issue with skin detection). Therefore, it may be necessary to set all pixel values that are below a noise threshold to a constant very small, non-zero value below the noise threshold value. It is important that the values be non-zero because of the considerations outlined in Section 2.7.3, Eqn. (2.29).

The input to the optional pre-processing stage is the raw multispectral or hyperspectral image cube ($\underline{\underline{X}}$). The output of the optional pre-processing stage is a similar image cube with altered pixel values ($\underline{\underline{X}}'$).

3.2 S2: Skin Detection Stage

The second stage of the proposed dismount detection system is the skin detection stage. The skin detection stage consists of three steps, as depicted in Fig. 3.3. The first step is to convert the input imagery to reflectance space using the empirical line method (ELM) as outlined in Section 2.7.1. The second step is to generate normalized difference skin index (NDSI) and either normalized difference vegetation index (NDVI) or normalized difference green-red index (NDGRI) features outlined in Section 2.7.3. The third stage is a skin-detection algorithm based on NDSI and either NDVI or NDGRI inputs.

The input to the skin detection stage is the raw or pre-processed multispectral or hyperspectral image cube ($\underline{\underline{X}}$ or $\underline{\underline{X}}'$). The output of the skin detection stage is a logical matrix of detected and rejected skin pixels ($\underline{Y}$).
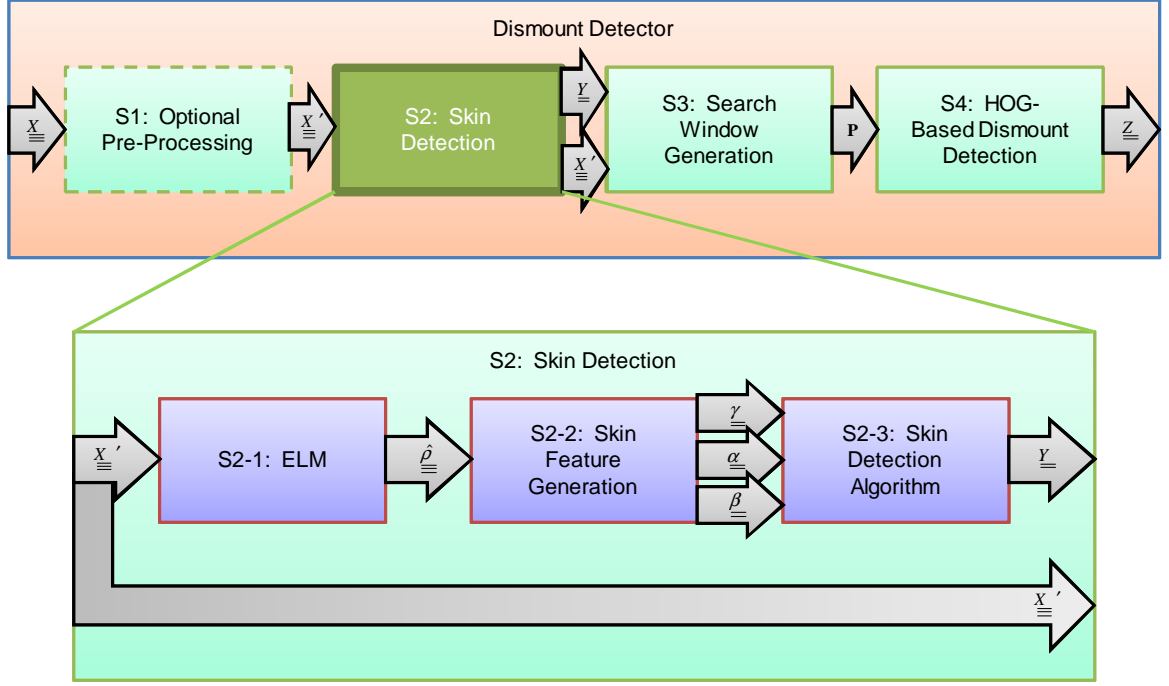
Figure 3.3:   Stage 2: Skin detection.

3.2.1 S2-1: Empirical Line Method Step.   In the ELM step, reflectance values are estimated for each image pixel at each wavelength of interest (540nm, 660nm, 1080nm, and 1580nm) using either two in-scene calibration targets of known reflectance with Eqn. (2.27) or one in-scene calibration target of known reflectance with Eqn. (2.28).

The input to the ELM step is the raw or pre-processed image cube ($\underline{\underline{X}}$ or $\underline{\underline{X}}'$). The output of the ELM step is a cube of estimated reflectance values ($\underline{\underline{\hat{\rho}}}$) with indices corresponding to each pixel in the original image cube.

3.2.2 S2-2: Skin Feature Generation Step.   During the skin feature generation step, NDSI, NDVI, and NDGRI features are generated for each pixel via Eqn. (2.31), Eqn. (2.32), and Eqn. (2.33) respectively using estimated reflectance values from the ELM step.

The input into the skin feature generation step is the estimated-reflectance cube ($\hat{\underline{\underline{\rho}}}$). The outputs of the skin feature generation step are matrices of NDSI, NDVI, and NDGRI values ($\underline{\underline{\gamma}}$, $\underline{\underline{\alpha}}$, and $\underline{\underline{\beta}}$ respectively) with indices corresponding to the original image pixel locations.

3.2.3 S2-3: Skin Detection Algorithm Step. There are numerous general detection techniques that can be applied to the skin detection problem. This thesis effort focuses on two general methods: the simple rules-based detector (Section 2.7.8 [51]) and a detector based on the likelihood-ratio test (LRT) (developed in this thesis).

The LRT from Section 2.8.1 is used to develop a LRT-based skin detection method. As discussed in Section 2.8.1, a two-dimensional likelihood ratio consisting of either a (NDVI,NDSI) or (NDGRI,NDSI) pair is generated as

$$S_i : i = \begin{cases} 1 & \text{if } \Lambda_\Theta\left(\theta\right) \equiv \frac{\hat{f}_1(\theta)}{\hat{f}_0(\theta)} > \eta_\Lambda \\ 0 & \text{if } \Lambda_\Theta\left(\theta\right) \equiv \frac{\hat{f}_1(\theta)}{\hat{f}_0(\theta)} < \eta_\Lambda \end{cases}, \tag{3.1}$$

where $\hat{f}_0(\theta) = P[\Theta = \theta | \text{not skin}]$, $\hat{f}_1(\theta) = P[\Theta = \theta | \text{skin}]$, $\Theta = \{\{A, \Gamma\}, \{B, \Gamma\}\}$, $\theta = \{\{\alpha, \gamma\}, \{\beta, \gamma\}\}$ are sets of parameters based on the (NDVI,NDSI) or (NDGRI,NDSI)-based detectors, $\hat{f}_1(\theta)$ is the estimated probability density function of human skin, $\hat{f}_0(\theta)$ is the estimated probability density function (pdf) of suspected false alarm sources.

The functional forms of $\hat{f}_1(\theta)$ and $\hat{f}_0(\theta)$ are estimated by Gaussian mixture models using Expectation Maximization [47] as described in Section 2.8.2 such that

$$\hat{f}_j\left(\theta\right) = \sum_{k=1}^{K_j} \pi_{j,k} \mathcal{N}\left(\underline{\mu}_{j,k}, \underline{\underline{\Sigma}}_{j,k}\right), j \in \{0, 1\}, \tag{3.2}$$

where K_j is the number of Gaussians utilized to estimate $\hat{f}_j\left(\theta\right)$, $\pi_{j,k}$ is the weighted value of each Gaussian such that $\pi_{j,k} \in \mathbb{R}[0, 1]$ and $\sum_{k=1}^{K_j} \pi_{j,k} = 1$. The parameters of

each Gaussian are represented by mean vector $\underline{\mu}_{j,k}$ and covariance matrix $\underline{\underline{\Sigma}}_{j,k}$. The likelihood ratio represents a two-dimensional decision surface.

The skin model described in Section 2.7.3 is used to generate samples to compute $\hat{f}_1(\theta)$. This makes the implicit assumption that all normal skin types are equally probable and that the specular reflection component is distributed uniformly with experimentally-determined upper and lower bounds ($c \sim \mathcal{U}[0.04, 0.14]$). The USGS spectral library [15] augmented with measurements with a hand-held spectrometer are used to generate $\hat{f}_0(\theta)$.

Once the functional forms of $\hat{f}_1(\theta)$ and $\hat{f}_0(\theta)$ are estimated, the likelihood ratio is computed and compared to the threshold η_Λ.

Inputs to the skin detection algorithm step are the NDSI, NDVI, and NDGRI values ($\underline{\underline{\gamma}}$, $\underline{\underline{\alpha}}$, and $\underline{\underline{\beta}}$ respectively) from the feature generation step. The output of the skin detection algorithm step is a logical matrix of detected and rejected skin pixels ($\underline{Y}$).

3.3   S3: Search Window Generation Stage

The third stage of the proposed dismount detection system is the search window generation stage. The search window generation stage consists of five steps, as depicted in Fig. 3.4. The first step is to label islands of contiguous skin-detection pixels. The second step is an optional processing step to reduce the number of skin-detection pixel islands that are of insignificant size. The third step is to calculate location properties of skin-detection pixel islands including centroids and extrema. The fourth step is to generate search windows based on the location properties of skin-detection pixel islands. The fifth step is to generate image patches from search windows determined by the previous step.

The inputs to the search window generation stage are a logical matrix of detected skin pixels ($\underline{Y}$) and the pre-processed image cube ($\underline{\underline{X}}'$). The output of the search window generation stage is a structure (**P**) of image patches corresponding to the
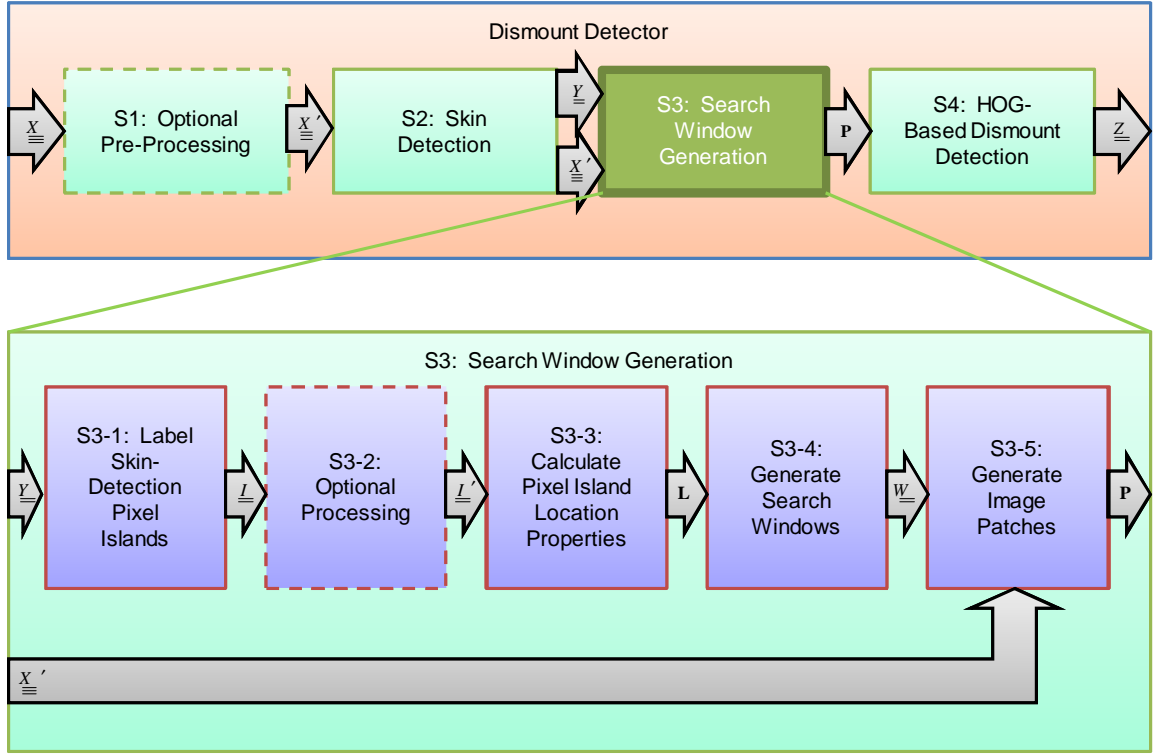
Figure 3.4:   Stage 3: Search window generation.

generated search windows and their corresponding bounding box coordinates in the
original image.

3.3.1   S3-1: Labeling Islands of Contiguous Skin-detection Pixels Step.   During the first step of the search window generation stage, islands of skin-detection pixels are given unique labels for further processing. MATLAB® provides the functions `bwlabel`, `bwlabeln`, and `bwconncomp` which automatically detect and label connected pixels as islands. The connection neighborhood (four nearest neighbors, eight nearest neighbors, etc.) is adjustable for each of the functions mentioned above. For the purpose of this thesis, the default neighborhood connectivity setting is eight-nearest-neighbors.

The input to the labeling islands of contiguous skin-detection pixels step is a logical matrix of detected skin pixels ($\underline{Y}$). The output of the labeling islands of

| (a) | (b) | (c) |
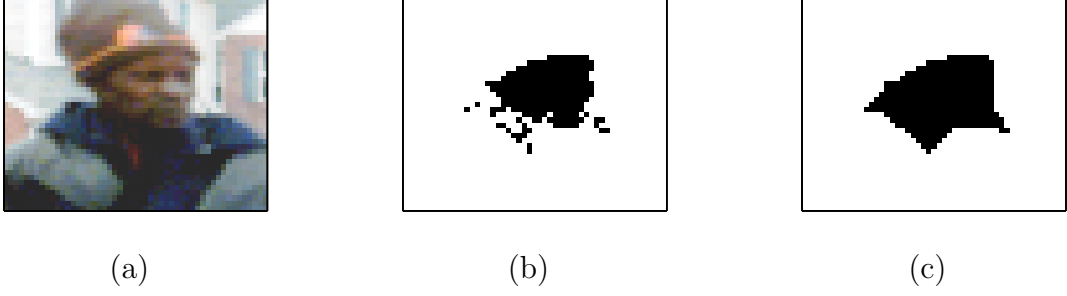
Figure 3.5:   Morphological close operation example. (a) Original image. (b) Original skin detection pixel islands. (c) Results of morphological close operation with $\delta = 8$.

contiguous skin-detection pixels step is a matrix of labeled skin-detection pixel islands ($\underline{\underline{I}}$).

3.3.2   S3-2: Skin-detection Pixel-island Processing Step.   During the optional skin-detection pixel-island processing step, morphological operations (such as a close operation with a disk structural element of radius δ, demonstrated in Fig. 3.5) and/or discarding islands with total pixels less than a threshold (η_A) can be useful for reducing pixel island edge artifacts and small "orphan" pixel islands. This may reduce the number of pixel islands, while raising the relative significance of each remaining pixel island.

The input to the skin-detection pixel-island processing step is a matrix of labeled skin-detection pixel islands ($\underline{\underline{I}}$). The output of the skin-detection pixel-island processing step is a similar matrix of labeled skin-detection pixel islands with possibly fewer, more-significant islands ($\underline{\underline{I}}'$).

3.3.3   S3-3: Skin-detection Pixel Island Location Properties Calculation Step. During the skin-detection pixel island location properties calculation step, the following properties are determined: the centroid and/or bounding extrema of each skin-detection pixel island. MATLAB® provides the function `regionprops` that efficiently provides this required information. Conveniently, the `regionprops` function accepts a labeled matrix of pixel islands (e.g. $\underline{\underline{I}}'$).

3-7

The input to the skin-detection pixel island location properties calculation step is a matrix of labeled pixel islands ($\underline{\underline{I}}$ or $\underline{\underline{I}}'$). The output of the skin-detection pixel island location properties calculation step is a structure of skin-detection pixel island location properties ($\mathbf{L}$).

3.3.4   S3-4: Search Window Generation Step.   During the search window generation step, skin-detection pixel island location properties are used to generate image patches of potential dismounts for later classification. Several approaches can be taken for generating search windows.

One search window generation approach is to generate windows surrounding each skin-detection pixel island ($\underline{\underline{I}}_i, i \in \mathbb{Z}[1, \xi]$ where ξ is the number of skin detection islands in the image) with every possible bounding box that contains $\underline{\underline{I}}_i$. In this approach, the sliding-window parameter set (θ_w as discussed in Section 2.6.1) is used to define search window shifting increments similar to the manner discussed in Section 2.6.1 and depicted in Fig. 3.6. This method makes no assumptions about the likely locations of skin within a search window. The advantage of this approach is that if there is any exposed skin on the dismount, a search window containing that dismount will be generated. The disadvantage of this approach is that a large set of search windows is generated, possibly negating much of the search-space reduction that could be provided by the system.

Another search window generation approach is to assume that the skin detections are limited to certain regions of a search window that positively contains a dismount. For example, if the assumption is made that all exposed skin is part of a face or head, only a relatively small set of search windows need be generated. The advantage of this approach is that significant search-space reduction can be realized. The disadvantage of this approach is that it limits the usefulness of exposed skin regions that are not in the assumed body locations. If there is no exposed skin in the assumed body locations, the dismount may not be detected, even though the dismount may have exposed skin in other locations. For example, if a dismount with
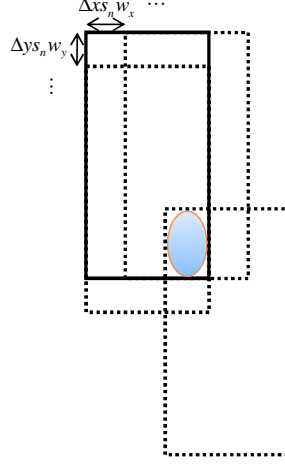
Figure 3.6:   Search window positioning relative to a skin detection pixel island with
no assumptions on skin position.

long hair is facing away from the camera but is wearing shorts, the skin detections in
the leg areas may not produce a set of search windows that would include the entire
dismount, while the long hair may obscure any skin in the head region, preventing
detection of the dismount.

For this thesis effort, it is assumed that all skin detections are in the face/head
region. Statistically, at least three out of four upright anatomical-plane aspects (front
coronal, back coronal, left sagittal, and right sagittal, as depicted in Fig. 3.7) of
the head will have exposed skin, logically making it the most likely body part to
have exposed skin visible to an imaging system. While there is a chance of missing
a detection, it is hypothesized that the impact to detection percentage is minimal
compared to the magnitude of the search space reduction.

In the worst-case scenario (i.e. all skin-detection pixel islands are perfectly
positioned such that the full range of scale values can be used), the number of search
windows produced for an $M \times N$ image using the face/head assumption is

$$\varsigma \leq \xi n_{\max} \left(1 + 2\zeta\right)^2, \tag{3.3}$$

3-9

Figure 3.7:   Anatomical-plane aspects of the human head in upright position. The top-left image is front coronal aspect. The top-right image is the back coronal aspect. The bottom-left image is the right-sagittal aspect. The bottom-right image is the left-sagittal aspect.

where ξ is the number of skin-detection pixel islands in the image, n_{max} is the maximum number of scales possible as determined by Eqn. (2.19) in Section 2.6.1, and ζ is a "slop" factor for generating additional search windows slightly offset from every search window cued to a skin-detection pixel island. Each slop window is offset by ζ increments of $\Delta x s_n w_x$ in the x-direction and $\Delta y s_n w_y$ in the y-direction. Adding slop windows may help account for how variations in skin-detection pixel island location statistics affect search window locations. For example, differences in hairline may affect centroid calculations for detected skin on the face, possibly affecting the position of the generated search window in relation to the rest of the body. Figure 3.8 depicts how ζ is utilized to generate additional slop windows to help mitigate such variations.
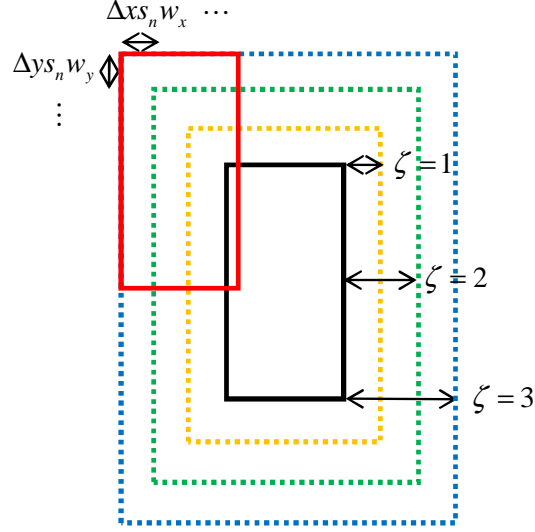
Figure 3.8: Additional "slop" search windows (red) are generated at intervals of $\Delta x s_n w_x$ and $\Delta y s_n w_y$ in the x and y directions respectively. The black window is the original search window. The dotted windows represent the limits, where all windows in between at intervals of $\Delta y s_n w_y$ and $\Delta x s_n w_x$ are also generated. The value of ζ determines how many intervals away from the original search window the slop-space should extend (orange for $\zeta = 1$, green for $\zeta = 2$, and blue for $\zeta = 3$.

There are multiple methods for determining where to position search windows relative to the location of $\underline{\underline{I}}_i$. One method is to position the windows relative to the centroid of $\underline{\underline{I}}_i$ centered in the x-direction with a scaled offset value $(s_n \Delta u)$ from the top of the window to the centroid of $\underline{\underline{I}}_i$ (Fig. 3.9 left). Another method is to position the windows centered in the x-direction based on the centroid of $\underline{\underline{I}}_i$ with a scaled offset value $(s_n \Delta v)$ from the top of the window to the top of $\underline{\underline{I}}_i$ (Fig. 3.9 right).

The advantage of the Δu-offset window positioning method is that it may be less prone to fluctuations in hairline/hat line. The advantage of the Δv-offset window positioning method is that it may be less prone to fluctuations in clothing in the neck and chest areas. Both methods are explored in this thesis.

Search windows are generated at each available scale in the sliding-window parameter set (θ_w). Additional windows offset in the x and y-directions may be gen-
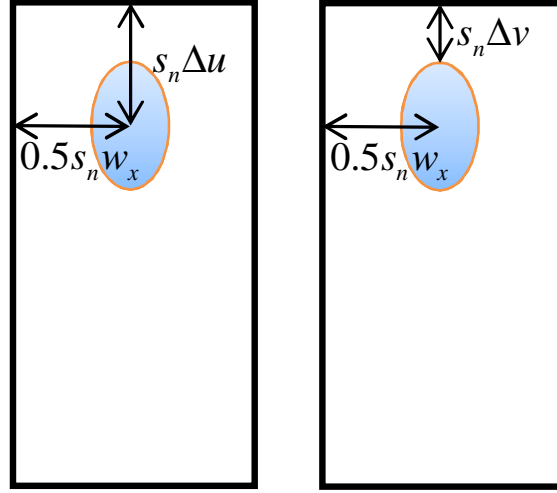
Figure 3.9:   Search window positioning relative to a skin detection pixel island. The left side of the figure illustrates the Δu method of positioning, while the right side of the figure illustrates the Δv method of positioning.

erated to account for variations in centroid locations due to shape, size, or aspect variations of skin-detection pixel islands.

The input to the search window generation step is a structure of skin-detection pixel island location properties ($\mathbf{L}$). The output of the search window generation step is a matrix of search-window bounding box coordinates ($\underline{\underline{W}}$).

3.3.5   S3-5: Image Patch Generation Step.   During the image patch generation step, image patches are extracted from the original or pre-processed image cube ($\underline{\underline{X}}$ or $\underline{\underline{X}}'$) for classification in the next stage. To generate each patch, image data within the spatial boundaries of each detector window bounding box is rescaled to the global detector window resolution (defined by w_x and w_y from the parameter set θ_w).

Rescaling the image data within each image patch is accomplished using bilinear interpolation (presented in Appendix A). The MATLAB® function `imresize` conveniently rescales an image from any arbitrary resolution to any arbitrary resolution
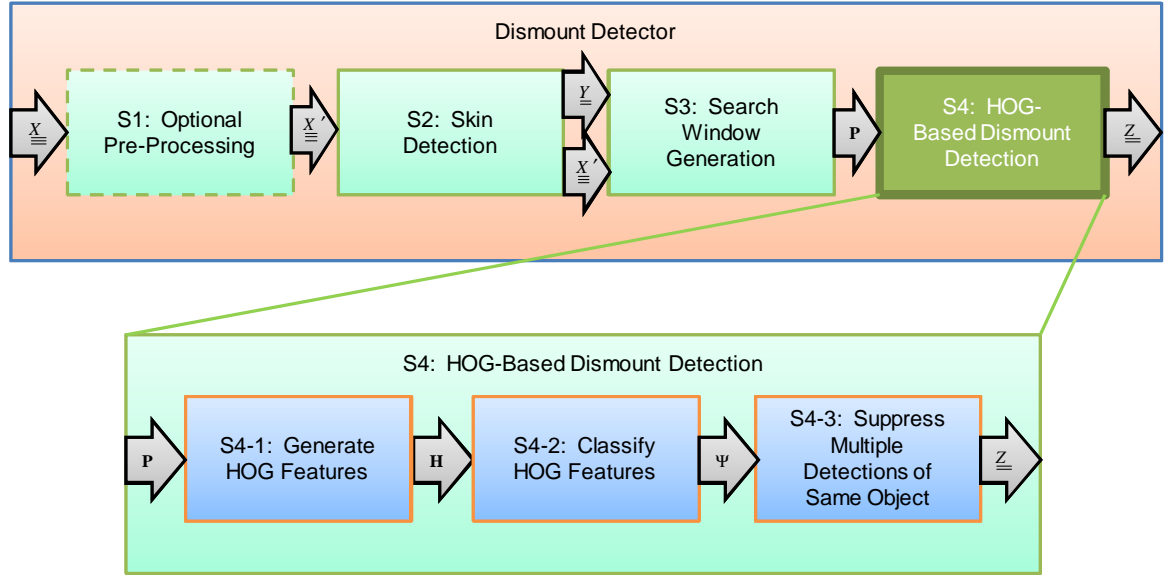
Figure 3.10:   Stage 4: HOG-based dismount detection.

with numerous options for calculating resultant pixel values (the default is bilinear interpolation).

The inputs to the image patch generation step are a matrix of search-window bounding box coordinates ($\underline{\underline{W}}$) and the image to which they apply ($\underline{\underline{X}}$ or $\underline{\underline{X}}'$). The output of the image patch generation stage is a structure of image patches and their corresponding bounding box coordinates in the original image ($\mathbf{P}$) ready for classification.

3.4   S4: HOG-based Dismount Detection Stage

The fourth stage of the proposed dismount detection system is the HOG-based dismount detection stage. The HOG-based dismount detection stage consists of three steps, as depicted in Fig. 3.10. The first step is to generate HOG features for each search window's corresponding image patch, as described in Section 2.5.1. The second step is to classify each resultant HOG feature. The third step is to suppress multiple detections of the same in-scene object so that only one detection per object exists.

The input to the HOG-based dismount detection stage is a structure of image patches and their corresponding bounding box coordinates in the original image ($\mathbf{P}$) ready for classification. The output of the HOG-based dismount detection stage is a matrix of dismount detection bounding box coordinates ($\underline{\underline{Z}}$).

3.4.1   S4-1: HOG Feature Generation.   During the HOG feature generation step, HOG features for each search window's corresponding image patch are generated as described in Section 2.5.1.

The input to the HOG feature generation step is a structure of image patches and their corresponding bounding box coordinates in the original image ($\mathbf{P}$) ready for classification. The output of the HOG feature generation step is a structure of HOG features and their corresponding bounding box coordinates in the original image ($\mathbf{H}$).

3.4.2   S4-2: HOG Feature Classification.   During the HOG feature classification step, HOG features corresponding to search windows are classified using linSVM as described in Section 2.5.2. The methodology for training the linSVM classifier employed in this thesis effort is provided in Section 4.5.2. The output confidence number from the linSVM is used to classify the HOG feature–and implicitly the search window it was generated from–as either a dismount or not a dismount by

$$S_i : i = \begin{cases} 1 & \text{if } \tau \leq \eta_\tau \\ 0 & \text{otherwise} \end{cases}, \tag{3.4}$$

where S_1 is the decision space where the classifier hypothesizes that the HOG feature is a dismount, S_0 is the decision space where the classifier hypothesizes that the HOG feature is not a dismount, τ is the prediction value provided by the linSVM, and η_τ is a detection threshold on the linSVM prediction value.

The input to the HOG feature classification step is a structure of HOG features and their corresponding bounding box coordinates in the original image ($\mathbf{H}$). The output of the HOG feature classification step is a structure of dismount detection

hypotheses and their corresponding bounding box coordinates in the original image ($\mathbf{\Psi}$).

3.4.3 *S4-3: Suppression of Multiple Detections of the Same Object.*

Since it is possible for several dismount detections to occur based on the same in-scene object (a side-effect of classifying at multiple scales and with minor spatial offsets), the suppression of multiple detections of the same object step utilizes confidence-based non-maximum suppression to reduce spurious detections, as described in Section 2.6.3.

The input into the suppression of multiple detections of the same object step is a structure of dismount detection hypotheses and their corresponding bounding box coordinates in the original image ($\mathbf{\Psi}$). The output of the suppression of multiple detections of the same object step is a matrix of unique dismount detection bounding box coordinates in the original image ($\underline{\underline{Z}}$).

3.5 Summary

This chapter provides methodology for using skin detections to cue a dismount detector based on HOG. The chapter begins by discussing data conditioning considerations, followed by a LRT-based skin detection algorithm. Next, considerations for how to position search windows relative to skin detections are presented. Finally, the HOG-based dismount detection process is presented.

IV. Experimental Results and Analyses

This chapter provides experimental procedures, experimental results, and analyses of results obtained by this thesis effort. Specifically, this chapter begins with descriptions of the data sets that are used. Next, skin feature trade-off studies and skin detector trade-off studies are performed, followed by a discussion of skin detection results.

Sliding-window detector scoring methodology and image truthing considerations are presented next, followed by validation of the results presented in [25]. The same search methodology used on the validation data is applied to a hyperspectral data set as a baseline for comparison between the full sliding-window histograms of oriented gradients (HOG)-based dismount detection scheme used in [25] and the skin-detection-cued HOG-based dismount detection scheme proposed by this thesis.

Next, trade-off studies of different skin detection search window cueing parameters are provided. Finally, the performance and search space requirements of the best skin-detection-cued dismount detector and the baseline dismount detector are compared.

4.1   Data Sets

Five different sources of data are used for different components of this research effort: two sets of hyperspectral reflectance measurements (data from the United States Geological Survey (USGS) [15] and a field spectrometer [1]), one set of modeled hyperspectral skin reflectance (from the model developed in [51], [55]), one set of hyperspectral imagery (from the HST3 imager [33]), and one set of panchromatic visible (VIS) imagery (from [25]).

4.1.1   United States Geological Survey Data Set.   The USGS Spectroscopy Lab has compiled an extensive library of spectral reflectance measurements [15]. Hundreds of materials have been measured and labeled, including 200 types of vegetation; 24 measurements of melting snow, seawater, and different concentrations of mud; and

1141 measurements of minerals, man-made materials, and chemicals. Measurements are provided from 0.2-15μm at varying sampling intervals (at or below 1nm). This research effort employs the USGS data set to train and test different skin detection algorithms in Section 4.2.

4.1.2 Field Spectrometer Data Set.   The USGS data set is augmented with measurements taken with an ASD FieldSpec® 3 portable spectrometer [1]. Included are 419 measurements of vegetation (heavily focused on the yew family since they are known false alarm sources [51], [52]); 110 measurements of melting snow, ice, murky water, and different concentrations of mud; and 250 measurements of other materials including soil, human hair of different colors, different types of stone, and feathers. Measurements are provided from 350-2500nm at 1-nm sampling intervals. This research effort employs these measured data in concert with the USGS data set to train and test different skin detection algorithms in Section 4.2.

4.1.3 Skin Reflectance Model Data Set.   The human skin reflectance model developed in [51], [55] is used to generate 3,936 unique samples of skin reflectance values $(\tilde{\rho}_\lambda^\perp)$ with a uniform distribution of all possible human skin parameters. In this way, the entire range of human skin types is represented in the data set, rather than being biased by the skin properties of available measurement subjects (which may not fully represent the possible range of skin properties, depending on demographics of the subject group). Modeled reflectance values are provided from 350-2500nm at 1-nm sampling intervals. This research effort employs these modeled data in concert with the USGS data set and field spectrometer data set to train and test different skin detection algorithms in Section 4.2.

4.1.4 Hyperspectral Data Set.   Hyperspectral imagery used for this research are collected with the SpecTIR HST3 Hyperspectral Imager [33]. The HST3 collects data in the range of 400-2500nm. The spectral bands are nominally 11nm wide in the VIS and 8nm wide in the near-infrared (NIR). The full width half maximum

(FWHM) of each of the bands is 14nm (VIS) and 8nm (NIR). Radiance spectra from the image cube are transformed into estimated reflectance using the ELM as described in Section 2.7.1.

Forty-two images are collected, including 39 images with one individual at varying distances from the camera at different times of day, and 4 images containing a large group of individuals at varying distances with varying skin colors from very light to very dark. All 42 images are used for dismount detection testing.

To test the skin detection algorithms, the 4 images containing many individuals are collected with skin color confusers and skin with various levels of pigmentation with a representative sample image in Fig. 4.1(top). Each of these 4 images contains typical color-based skin detection confusers to include a flesh-colored doll, a piece of cardboard, and a red brick. Other color confusers include a leather boot and several pieces of wood. These objects are selected because their colors are similar to some shades of skin [51], [53]. A branch from a conifer (from the yew family) is included in the scene as it tends to have a high NDSI value. The scene is a suburban environment with houses, streets, sidewalks, trees, typical yards with grass, bushes, bark, and other assorted materials. Portions of the reference panels in scene are used to estimate reflectance using the ELM and are visible in the bottom right portion of the figure. Fig. 4.1(bottom) shows the corresponding skin truth mask.

4.1.5 Daimler Benchmark Data Set. The Daimler Benchmark data set is a collection of panchromatic VIS imagery provided by [25]. The data set includes 15,660 dismount image patches for positive training, 6,744 full images containing no dismounts for negative training, and 21,790 test images including truth window locations for in-scene dismounts. All of the Daimler Benchmark training data are used to train the HOG-based dismount detector. A random subset of 264 images from the Daimler Benchmark test image suite are used to validate the HOG-based dismount detector performance due to computational time constraints.
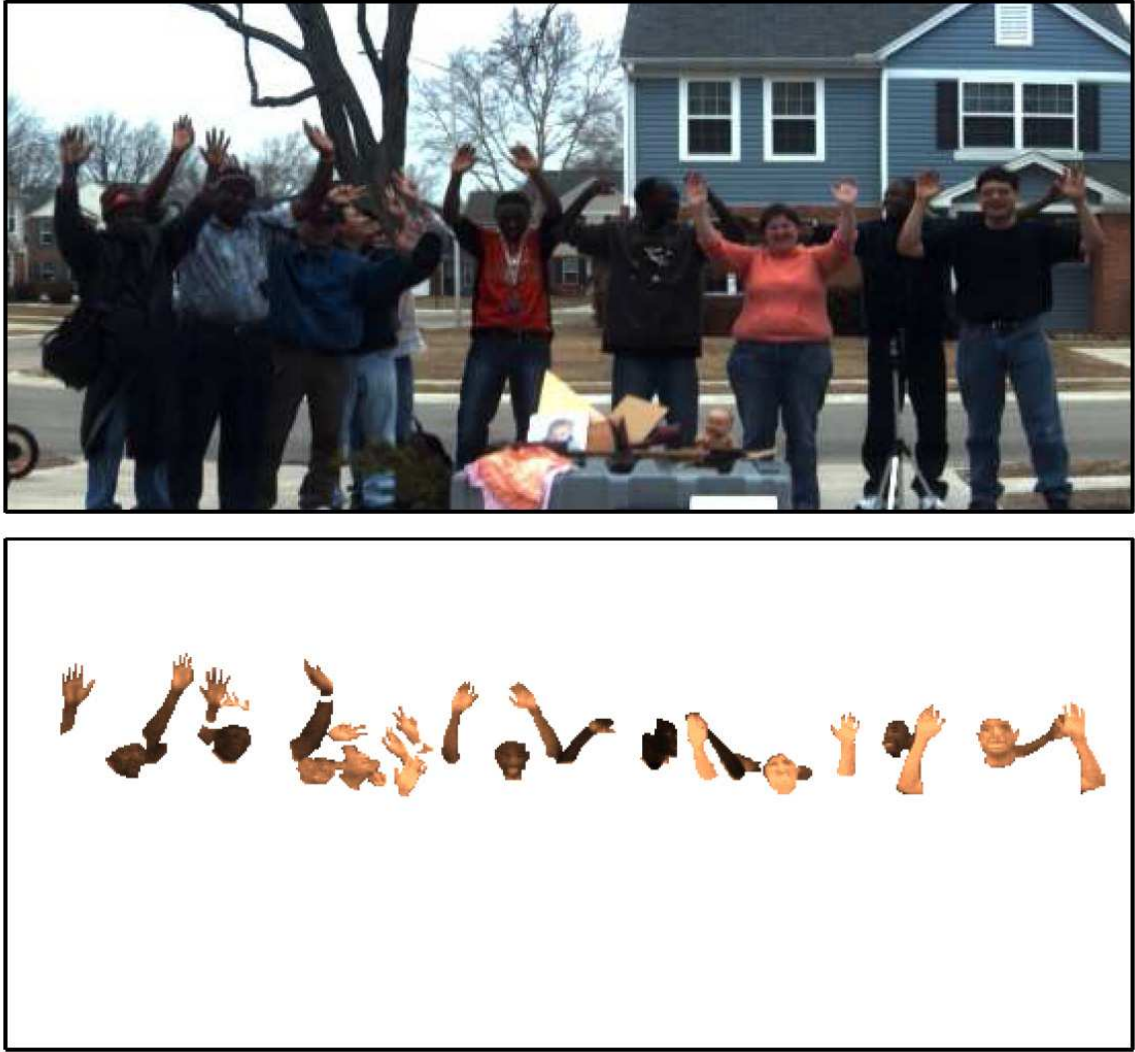
Figure 4.1:   Skin truth HyperSpecTIR version 3 (HST3) image. Color image of
suburban test scene (top) and the skin truth pixels (bottom). The scene
contains people with different skin colors as well as several potential
false alarm sources.

4.2   Skin Detection: Considerations and Results

Table 4.1 provides a list of NDSI, NDGRI, and NDVI values for different materials including skin with different pigmentation levels, skin confusers, and typical background material in a rural scene. As one would anticipate, materials with significant water content, such as vegetation and skin, have the highest NDSI values. Also note that the NDSI values for the darkest skin *can* be higher than values for

vegetation (i.e., separability between skin and vegetation in NDSI space is possible, but is not guaranteed). Vegetation has the highest NDVI values and objects that are green have the highest NDGRI values.

Table 4.1:   NDVI, NDSI, and NDGRI values for different materials.

Material	NDVI	NDSI	NDGRI
Fair Skin	0.04	0.77	-0.25
Dark Skin	0.51	0.66	-0.34
Paper Bag	0.27	0.15	-0.27
Cardboard	0.3	0.14	-0.33
Red Brick	-0.01	-0.01	-0.47
Salt Water	-0.10	0.02	0.20
Muddy Water	0.04	0.85	-0.10
Grass	0.88	0.53	0.37
Leaf	0.9	0.27	0.41
Doll	0.04	0.24	-0.28
Soil	0.37	-0.1	-0.18
Mud	0.21	-0.18	-0.20
Snow	-0.19	0.93	0.01
Conifer	0.83	0.40	0.47

Data used to generate the scatter plots in Fig. 4.2 are obtained from the USGS spectral library [15] and reflectometer measurements of known false alarm sources and skin from living subjects and cadavers as well as model generated data spanning the possible skin reflectance of normal human skin. False alarm sources include vegetation such as conifers and heavy water containing substances that are highly forward scatter such as snow, salt water, crushed ice, and liquid water with suspended materials (such as silt and sand). A two-dimensional scatter plot of the (NDVI,NDSI) pair is shown in Fig. 4.2 (left) and the (NDGRI,NDSI) pair in Fig. 4.2 (right).

From Table 4.1 and Fig. 4.2, it is clear that either the NDVI or NDGRI can be used to suppress false alarms when used in conjunction with the NDSI to identify skin. If one is searching for fair to moderately-pigmented persons in a scene with a

| | (a) (NDVI,NDSI) pair | (b) (NDGRI,NDSI) pair |

Figure 4.2:   (a) Joint distribution of NDVI and NDSI values using spectral measurements, skin model generated data, and living and cadaver skin data. (b) Joint distribution of NDGRI and NDSI values using spectral measurements, skin model generated data, and living and cadaver skin data. Spectral skin confuser measurements are shown as red circles, skin model generated data are shown as black dots, and skin measurements (living and cadaver) are shown as green '+'.

significant amount of vegetation, the NDVI algorithm may be an effective method for filtering out water-rich vegetation. However, darkly pigmented people have a high NDVI value and may be incorrectly discarded by a NDVI threshold set too low. If one is searching for people in an urban environment, the NDGRI can filter out pixels that are more green than red in a scene. However, the NDGRI would have greater difficulty identifying vegetation under low signal-to-noise ratio conditions (observe from Table 4.1 that NDVI>NDGRI for vegetation). The use of NDVI and NDGRI in suppression of false alarms when combined with the NDSI for skin detection is explored in the following sections. Specifically, this section presents a simple rules-based detection scheme and a LRT-based detection scheme and demonstrates the differences in false alarm suppression using both the NDVI and NDGRI.

4.3   Skin Detection Results for Modeled Data

To get an idea of performance in a controlled environment with the most diverse data set available, the rules-based and likelihood-ratio test (LRT)-based skin detec-

Table 4.2:   Noise variance as a function of wavelength. Variances are computed from reflectance measurements obtained from the SpecTIR HST3 Hyperspectral Imager [33]. Values are reported as 10^{-4}.

Wavelength	540nm	660nm	750nm	850nm
Variance	6.69	6.09	7.16	6.68
Wavelength	860nm	1080nm	1580nm	
Variance	6.71	8.29	9.01	

tors are tested on the combination of modeled human skin data and the USGS spectral library [15] data augmented with field samples collected using a hand-held spectrometer. Modeled skin data are modified as described earlier using the signal-plus-noise model described in Eqn. (2.34) with noise parameters described in Table 4.2. USGS spectral library and field sample data are modified with the estimated sensor noise only.

In order to test skin detection algorithms on modeled data, it is useful to simulate both sensor noise and specular reflections as described in Eqn. 2.36. Although [41] provides measurements of the specular component of human skin, these values are measured for broad-band energy and not as a function of wavelength. Furthermore, there is no translation for this work to map similar measurements in radiance to reflectance. As such, observation of the hyperspectral data from the HST3 sensor is used to estimate reasonable specular components where it is assumed that the specular component *is not* wavelength-dependent. The sensor noise component is spectrometer-dependent and is assumed to be the noise term in estimated reflectance (that is, after atmospheric correction). In the case of the HST3 system, there are two spectrometers (one VIS and one NIR).

Adding uniform distributed specular reflection of $c \sim \mathcal{U}[0.04, 0.14]$ (at 0.05 intervals) and sensor noise described in Table 4.2 to the (NDVI,NDSI) and (NDGRI,NDSI) pairs from Fig. 4.2 are shown in Fig. 4.3. Although specular reflection is assumed constant as a function of wavelength, it does exhibit spatial variation. The spatial distribution of specular reflections highly depends on the illumination angle (includ-

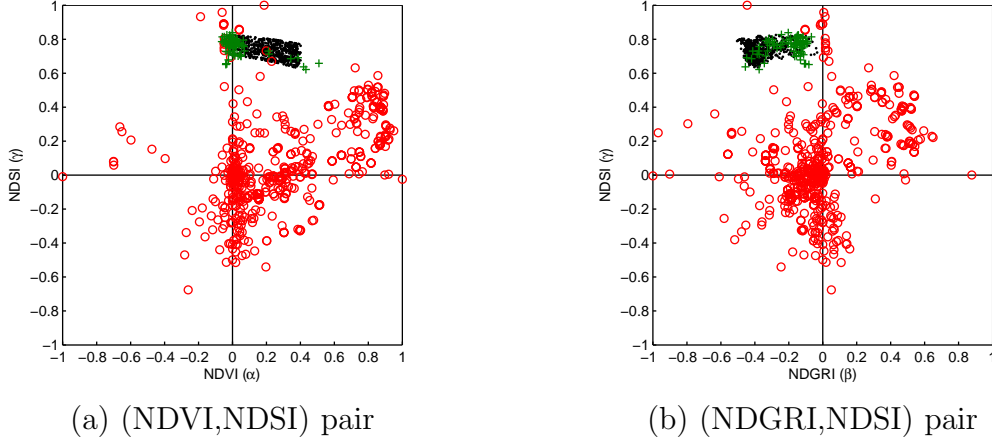<div align="center">(a) (NDVI,NDSI) pair   (b) (NDGRI,NDSI) pair</div>
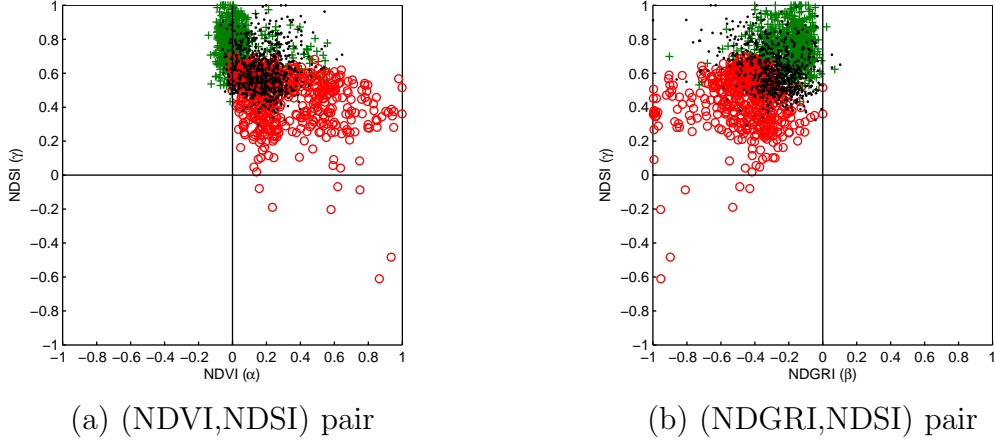
Figure 4.3:   (a) Joint distribution of NDVI and NDSI values using spectral measurements, skin model generated data, and living and cadaver skin data. (b) Joint distribution of NDGRI and NDSI values using spectral measurements, skin model generated data, and living and cadaver skin data. HST3 imaged skin data are shown as red circles, skin model generated data are shown as black dots, and skin measurements (living and cadaver) are shown as green '+'.

ing secondary illumination sources such as reflections from buildings), the observation angle, and the subject's surface geometry. The number of additional noisy samples added to the detector model is based on the desired distribution of specular reflections. In this way, it is possible to add an appropriate amount of noise to simulate any sensor's response while also accounting for varying percentages of specular reflections in the scene. Since the true distribution of the data is unknown, the distribution with the most uncertainty (i.e. the uniform distribution) is used to model the data for samples shown in Fig. 4.3 (as dictated by information theory). Visually comparing the distribution of NDGRI and NDSI skin values generated signal-plus-noise model in Eqn. 2.36 (depicted as black dots in Fig. 4.3) to (NDGRI,NDSI) pairs observed from the HST3 system (depicted as red circles in Fig. 4.3) visually shows a reasonable match.

The results presented in Fig. 4.4 and summarized in Table 4.3 and Table 4.4 are an aggregate of 20 noise realizations where each noise realization is further subject to K-Fold cross validation (for K=5) for each noise realization. The average performing

ROC curve is the mean of the 100 simulations (5 cross validation runs × 20 noise realizations).

Results of the detectors are presented as ROC curves in Fig. 4.4. The rules-based detectors for the (NDVI,NDSI) and (NDGRI,NDSI) pairs are presented in Fig. 4.4(a) and (b) respectively where the boundary values for the NDVI are $\alpha \in \mathbb{R}[-1, \{0.5, 0.6, 0.7, 1\}]$, the NDGRI are $\beta \in \mathbb{R}[-1, \{-0.02, -0.05, 0.1, 1\}]$, and the NDSI detector lower boundary varies as $\gamma \in \mathbb{R}[\{\mathbb{R}[-1, 0.93]\}, 0.93]$ (where 0.93 is an experimentally determined upper bound). (Note that for the $\alpha \in \mathbb{R}[-1, 1]$ and the $\beta \in \mathbb{R}[-1, 1]$ cases, the detector becomes an NDSI-based detector only and provides a baseline for comparison between NDVI and NDGRI-based detector performances). Results of the LRT-based detector for the (NDVI,NDSI) and (NDGRI,NDSI) pairs are presented in Fig. 4.4(c) and (d) respectively.

In both the rules-based and LRT-based detectors, the (NDGRI,NDSI) feature pair performs better then the (NDVI,NDSI) feature pair. The rules-based detector performs on average better than the LRT detector. However, when considering best case performance, the LRT outperforms the rules-based detector.

One should note that neither the rules-based nor the LRT detector ROC curves are strictly concave down. In the rules-based detector case, this is likely due to the fact that it is not optimal for minimizing the Bayes risk. In the LRT detector case, this is likely due to our assumption that a GMM adequately represents the true distribution of target and non-target samples when in fact this assumption does not hold true.

The error bars depicted in Fig. 4.4 represent the average $\pm$ standard deviation in the P_D and P_{FA} directions respectively. This is done at arbitrary points along each average ROC curve to illustrate the performance envelope. In general, variance in the P_{FA} direction is worse than in the P_D direction. This is intuitive since there is more variation in the non-skin class (i.e., the entire universe that is not skin) than the skin class.

The P_D and P_{FA} variance is greater for the LRT detectors than for the rules-based detectors because for each fold in the K-fold cross validation, a new LRT detector is computed. This is important to note because while the purpose of cross-validation is to attempt to remove bias when assessing performance, it comes at the cost of increased variance of the results [31]. Conversely, the rules-based detector does not change between folds, only the test set it is applied to.

Specific operating points (OPs) drawn from the ROC curves in Fig. 4.4 for a constant $P_{FA} = 0.0005$ and constant $P_D = 0.95$ are shown in Table 4.3 and Table 4.4. Complimentary OPs are provided for the minimum, average, and maximum values attained for the best average performing ROC curve. For the rules-based skin detector, the best average performing curve over one of four detector regions is considered. Each rules-based skin detector region is specified by upper and lower bounds on the NDVI or NDGRI thresholds ($\alpha \in [a_1, a_2]$ and $\beta \in [b_1, b_2]$). The lower NDSI threshold, c_1, is varied over the range $\mathbb{R}[-1, 0.93]$ (where $c_2 = 0.93$ is an experimentally determined upper bound). For the LRT-based skin detector, the average of all 100 results is used where models are recomputed for each fold in the cross validation for each of the noise realizations.

The summaries in Table 4.3 and Table 4.4 indicate that for a $P_D = 0.95$, NDVI results in a higher false alarm rate than does NDGRI. This is the case for both the rules-based and LRT-based skin detectors. At that specified operating point, the rules-based and LRT-based skin detectors perform in a similar manner with the exception of the maximum error where the rules-based has a lower P_{FA}.

For a $P_{FA} = 0.0005$, the rules-based detector consistently produces a higher P_D for NDGRI versus NDVI. In the case of the LRT detector, the best performing case for the NDGRI outperforms the NDVI, and by default this is true for the minimum performance since NDVI does not have a defined P_D at this operating point and the NDGRI does.

Table 4.3: Summary of the rules-based skin detector results for the modeled skin samples and reflectometer measurements of false alarm sources.

Feature	Operating Point (OP)	Complementary OP: Min/Avg/Max	Detector Param1	Detector Param2
NDVI	$P_D = 0.95$	$P_{FA} = 0.014/0.015/0.016$	$a_1 = -1.000$	$c_1 = 0.400$
			$a_2 = 0.500$	$c_2 = 0.930$
NDVI	$P_{FA} = 0.0005$	$P_D = 0.003/0.011/0.018$	$a_1 = -1.000$	$c_1 = 0.900$
			$a_2 = 0.600$	$c_2 = 0.930$
NDGRI	$P_D = 0.95$	$P_{FA} = 0.007/0.008/0.009$	$b_1 = -1.000$	$c_1 = 0.380$
			$b_2 = -0.05$	$c_2 = 0.930$
NDGRI	$P_{FA} = 0.0005$	$P_D = 0.022/0.046/0.119$	$b_1 = -1.000$	$c_1 = 0.860$
			$b_2 = -0.050$	$c_2 = 0.930$

Table 4.4: Summary of the LRT-based skin detector results for the modeled skin samples and reflectometer measurements of false alarm sources.

Feature	Operating Point (OP)	Complementary OP: Min/Avg/Max	Detector Param
NDVI	$P_D = 0.95$	$P_{FA} = 0.009/0.014/0.021$	$\eta_\Lambda = 3.000/38.000$
NDVI	$P_{FA} = 0.0005$	$P_D = \text{NA}/0.003/0.211$	$\eta_\Lambda = \text{NA}/187.000$
NDGRI	$P_D = 0.95$	$P_{FA} = 0.008/0.009/0.014$	$\eta_\Lambda = 4.000/8.000$
NDGRI	$P_{FA} = 0.0005$	$P_D = 0.000/1.36 \times 10^{-5}/0.297$	$\eta_\Lambda = 1.05 \times 10^5/40.000$

(a) (NDVI,NDSI) Rules-based method   (b) (NDGRI,NDSI) Rules-based method

(c) (NDVI,NDSI) LRT-based method   (d) (NDGRI,NDSI) LRT-based method
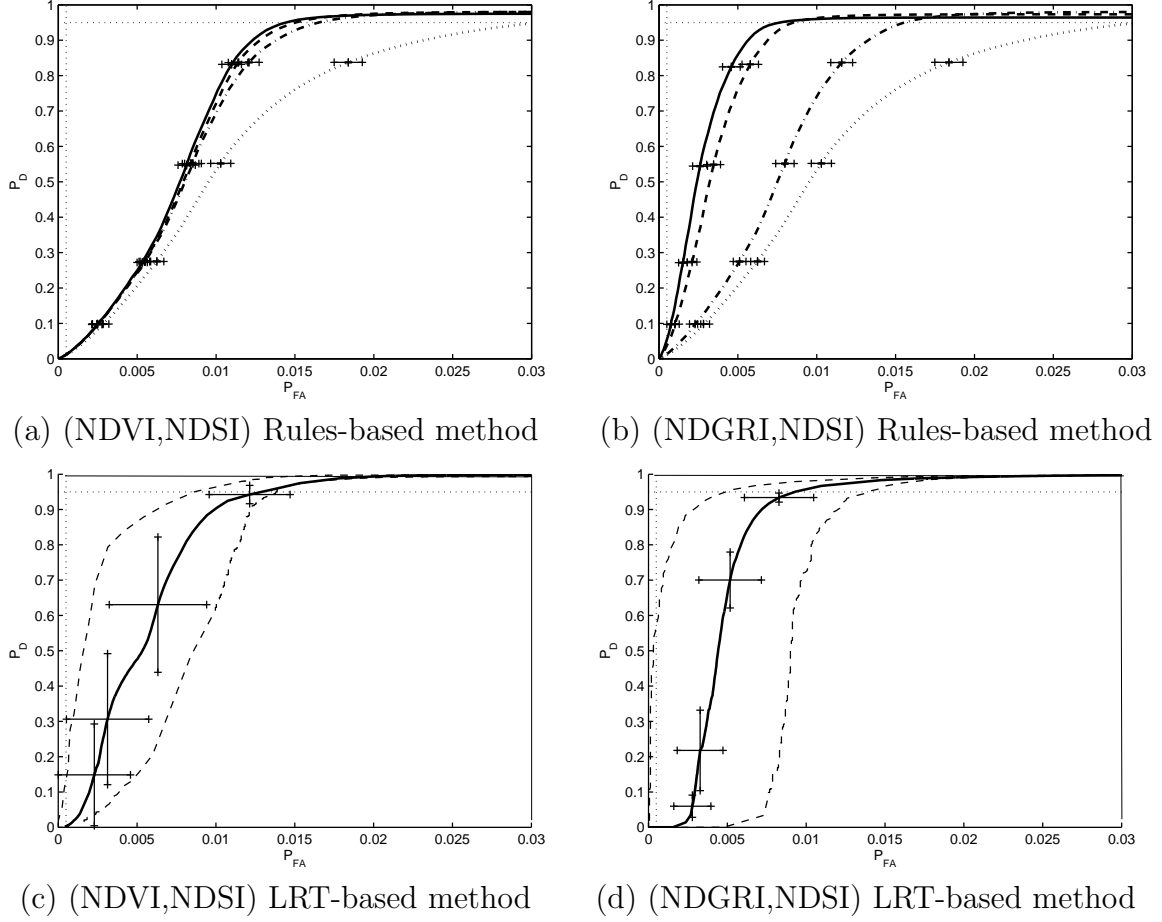
Figure 4.4:   The receiver operating characteristic (ROC) curves in (a)-(d) are for the modeled skin data and spectral library false alarm source data. The vertical dashed line represents a constant $P_{FA} = 0.0005$ while the horizontal dashed line represents a constant $P_D = 0.95$. (a) ROC curve for (NDVI,NDSI) pair using the rules based detector varying the lower bound of $\gamma \in \mathbb{R}[c_1 \in \mathbb{R}[-1, 0.92], 0.93]$ fixing the upper bound on NDVI $\alpha \in \mathbb{R}[-1, \{0.05, 0.06, 0.07, 1.0\}]$ (solid, dashed, dash-dotted, and dotted curves) yielding four detector regions. (b) ROC curve for (NDGRI,NDSI) pair using the rules based detector varying the lower bound of $\gamma \in \mathbb{R}[c_1 \in \mathbb{R}[-1, 0.92], 0.93]$ fixing the upper bound on NDVI $\beta \in \mathbb{R}[-1, \{-0.02, -0.05, -0.1, 1.0\}]$ (solid, dashed, dash-dotted, and dotted curves) yielding four detector regions (c) ROC curve for (NDVI,NDSI) pair using the LRT based detector varying $\eta_\Lambda \in \mathbb{R}[0, 5 \times 10^6]$. (d) ROC curve for (NDGRI,NDSI) pair using the LRT detector varying $\eta_\Lambda \in \mathbb{R}[0, 5 \times 10^6]$.

4.3.1 Skin Detection Results for Hyperspectral Test Imagery.   Due to the noise inherent in the system/environment and the fact that the bands selected for

Table 4.5:   HyperSpecTIR version 3 (HST3) bands used to implement skin detection algorithms.

Target λ	Band 1	Band 2	Band 3
540nm	531.37nm	542.74nm	554.08nm
660nm	648.68nm	660.27nm	672.00nm
750nm	743.14nm	754.70nm	766.49nm
850nm	837.50nm	849.05nm	860.89nm
860nm	849.05nm	860.89nm	872.77nm
1080nm	1069.91nm	1078.06nm	1086.29nm
1580nm	1570.83nm	1579.03nm	1587.27nm

skin detection algorithms do not line up with the HST3 band centers, the NDVI, NDSI, and NDGRI algorithms are modified to accommodate the available spectra. The algorithms are implemented with the mean of the estimated reflectance of the three HST3 bands closest to the algorithms' band centers, which helps suppress sensor noise. For example, the estimated reflectance at 540nm used for the NDGRI algorithm is implemented using the mean of the estimated reflectance from the HST3 bands centered at 531.37nm, 542.74nm, and 554.06nm. The band centers for the HST3 estimated reflectance that correspond to the band centers of the algorithm described earlier are provided in Table 4.5.

The ROC curves for the rules-based and LRT-based skin detectors on the hyperspectral image data are presented in Fig. 4.5. Note that in the case of the image data, ROC curves are concave down. For the rules-based detector, the same four detector regions used in Section 4.2 are used to generate the detection results on the hyperspectral image data. Similarly, the 100 detectors used to generate the detector results for the LRT detector described in Section 4.2 are used on the hyperspectral image data.

As noted previously, using the NDVI in both detectors produces the worst results. The disparity between the NDVI and NDGRI methods on the rules-based detector is significant. This is not so in the case of the LRT-based skin detector,

although there is clear performance gain using the NDGRI over the NDVI. Overall, the rules-based detector outperforms the LRT detector for the image data. This may be attributed to one of several reasons: fewer false alarm types exist in the image data versus the spectral library data; a bias in the skin reflectance model that works favorably on the image data; the rules-based method is tuned to the hyperspectral image data.

Consistent with the previous analysis, specific OPs drawn from the ROC curves in Fig. 4.5 for a constant $P_{FA} = 0.0005$ and constant $P_D = 0.95$ are shown in Table 4.6 and Table 4.7. Complimentary OPs are provided for the minimum, average, and maximum values attained for the best average performing ROC curve. For the rules-based detector, the best average performing curve over one of four detector regions is considered where each detector region is specified by upper and lower bounds on the NDVI or NDGRI thresholds ($\alpha \in [a_1, a_2]$ and $\beta \in [b_1, b_2]$). The lower NDSI threshold, c_1, is varied over the range $\mathbb{R}[-1, 0.93]$ (where $c_2 = 0.93$ is an experimentally determined upper bound). For the LRT detector, the average of all 100 results is used where models are recomputed for each fold in the cross validation for each of the noise realizations.

The summaries in Table 4.6 and Table 4.7 indicate that for a $P_D = 0.95$, NDVI results in a higher false alarm rate than does NDGRI. This is the case for both the rules-based and LRT detectors. At that specified operating point, the rules-based and LRT detectors perform in a similar manner with the exception of the maximum error where the rules-based skin detector has a lower P_{FA}. For a $P_{FA} = 0.0005$, the rules-based skin detector consistently produces a higher P_D for NDGRI versus NDVI.

4.3.1.1 Skin Detection Discussion. Two important results are evident in the skin detector outcomes. First, NDGRI appears to better suppress false alarms compared to the NDVI. This is intuitive since the false alarm sources in general are more green than they are red. Second, the rules-based skin detection method compares

Table 4.6:   Summary of the rules-based skin detector results for the HST3 image data.

Feature	Point of Interest (POI)	Complementary POI: Min/Avg/Max	Detector Param1	Detector Param2
NDVI	$P_D = 0.95$	$P_{FA} = 0.016/0.016/0.016$	$a_1 = -1.000$ $a_2 = 1.000$	$c_1 = -1.000$ $c_2 = 0.930$
NDVI	$P_{FA} = 0.0005$	$P_D = 0.760/0.760/0.760$	$a_1 = -1.000$ $a_2 = 0.700$	$c_1 = 0.420$ $c_2 = 0.930$
NDGRI	$P_D = 0.95$	$P_{FA} = 0.004/0.004/0.004$	$b_1 = -1.000$ $b_2 = -0.020$	$c_1 = 0.260$ $c_2 = 0.930$
NDGRI	$P_{FA} = 0.0005$	$P_D = 0.820/0.820/0.820$	$b_1 = -1.000$ $b_2 = -0.020$	$c_1 = 0.410$ $c_2 = 0.930$

Table 4.7:   Summary of the LRT-based skin detector results for the HST3 image data.

Feature	Point of Interest (POI)	Complementary POI: Min/Avg/Max	Detector Threshold
NDVI	$P_D = 0.95$	$P_{FA} = 1.000/1.000/1.000$	$\eta_\Lambda = 0.000/0.000$
NDVI	$P_{FA} = 0.0005$	$P_D = 0.662/0.669/0.689$	$\eta_\Lambda = 3.000/2.000$
NDGRI	$P_D = 0.95$	$P_{FA} = 0.004/0.004/0.005$	$\eta_\Lambda = 0.034/0.022$
NDGRI	$P_{FA} = 0.0005$	$P_D = 0.772/0.776/0.788$	$\eta_\Lambda = 3.000/4.000$

(a) (NDVI,NDSI) Rules-based method   (b) (NDGRI,NDSI) Rules-based method

(c) (NDVI,NDSI) LRT-based method   (d) (NDGRI,NDSI) LRT-based method
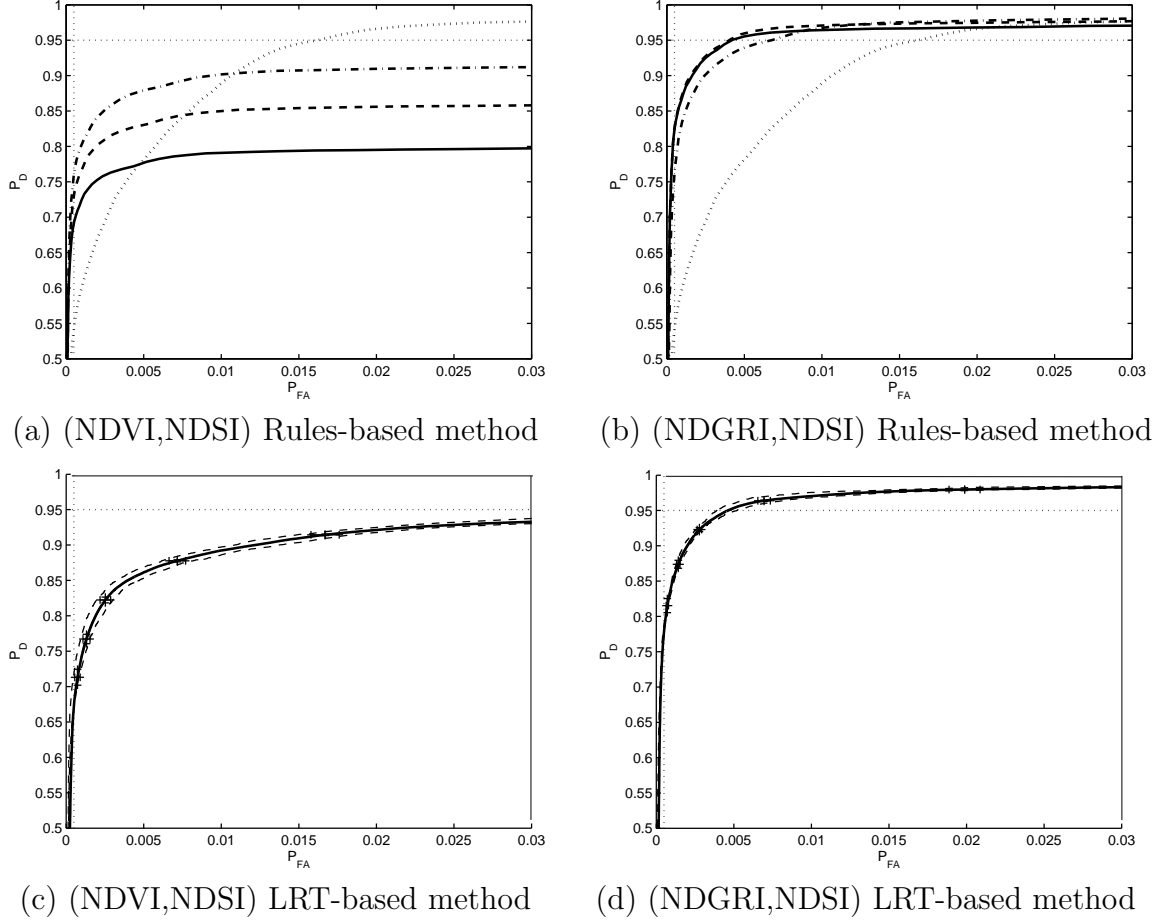
Figure 4.5:   The ROC curves in (a)-(d) are for a set of hyperspectral images similar to that of Fig. 4.1(top). The vertical dashed line represents a constant $P_{FA} = 0.0005$ while the horizontal dashed line represents a constant $P_D = 0.95$. (a) ROC curve for (NDVI,NDSI) pair using the rules based detector varying the lower bound of $\gamma \in \mathbb{R}[c_1 \in \mathbb{R}[-1, 0.92], 0.93]$ fixing the upper bound on NDVI $\alpha \in \mathbb{R}[-1, \{0.05, 0.06, 0.07, 1.0\}]$ (solid, dashed, dash-dotted, and dotted curves) yielding four detector regions. (b) ROC curve for (NDGRI,NDSI) pair using the rules based detector varying the lower bound of $\gamma \in \mathbb{R}[c_1 \in \mathbb{R}[-1, 0.92], 0.93]$ fixing the upper bound on NDVI $\beta \in \mathbb{R}[-1, \{-0.02, -0.05, -0.1, 1.0\}]$ (solid, dashed, dash-dotted, and dotted curves) yielding four detector regions (c) ROC curve for (NDVI,NDSI) pair using the LRT based detector varying $\eta_\Lambda \in \mathbb{R}[0, 5 \times 10^6]$. (d) ROC curve for (NDGRI,NDSI) pair using the LRT detector varying $\eta_\Lambda \in \mathbb{R}[0, 5 \times 10^6]$.

favorably with the LRT-based skin detection method, which comes at somewhat of a surprise since there is no optimality criterion in the rules-based detector.

The skin detection algorithm used for the remainder of this thesis effort is the rules-based detector with parameters $\beta \in [-1, -0.02]$, $\gamma \in 0.26, 0.93]$. The rules-based detector is chosen for computational efficiency and parameter adjustability.

4.4 Search Window Generation Results

One of the primary goals of this research effort is to significantly reduce the search space for a HOG-based dismount detector. Table 4.8 lists the maximum number of search windows that can be generated for several image sizes using either the full search space or skin-detection-cued search space. Equations (2.23) and (3.3) with the sliding window parameter set

$$\theta_w = \{w_x = 48, w_y = 96, h_{\min} = 72, w_{\min} = 0, \Delta s = 1.1, \Delta x = 0.1, \Delta y = 0.025\},$$

are used to calculate values for Table 4.8.

Table 4.8: Maximum number of search windows possible by image size where ξ is the number of skin detection pixel islands and ζ is the slop factor, as described in Section 3.3.4.

Image Size	Full Search	Skin-detection-cued Search			
		$\zeta = 0$	$\zeta = 1$	$\zeta = 2$	$\zeta = 3$
640×480	1.85×10^5	20ξ	180ξ	500ξ	980ξ
640×512	2.01×10^5	21ξ	189ξ	525ξ	1029ξ
1080×250	1.22×10^5	14ξ	126ξ	350ξ	686ξ

As Table 4.8 indicates, the number of search windows generated using the skin-detection-cueing approach can be orders of magnitude smaller than the full number of search windows generated from the sliding window parameter set θ_w, depending on the number of skin-detection pixel islands present in the image. Figure 4.6 illustrates the maximum number of possible search windows for a 1080×250-pixel image as a function of the number of skin detection pixel islands (ξ).

Frequently, the skin detection algorithm produces several very small pixel islands that are near other, larger pixels islands. This may be the result of sensor
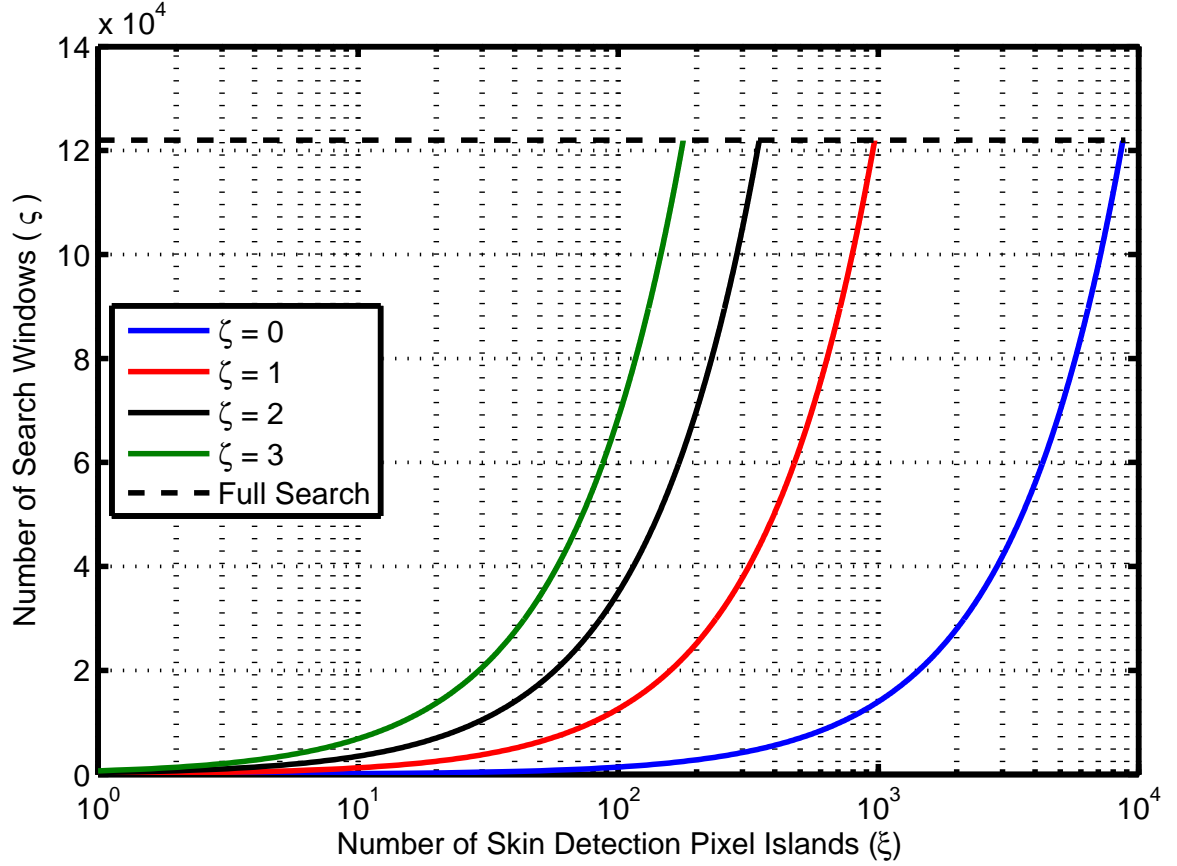
Figure 4.6:   Maximum number of search windows possible for a 1080×250-pixel image.

noise, surface geometry near an edge, mixed pixels, false alarm sources, etc. Eliminating skin-detection pixel islands that are less than a certain size may significantly reduce the number of pixel islands ξ and therefore the number of search windows per Table 4.8.

One method of reducing search windows is to attempt to merge smaller skin-detection pixel islands with other skin-detection pixel islands nearby. To test this, a morphological close operation using a disk structural element with radius (δ) varying from 0 to 20 is used to merge nearby skin-detection pixel islands together. Figure 4.7 depicts how varying the radius of the disk structural element used in the close operation affects the number of search windows produced for the entire HST3 data set of 42 images. For simplicity, all images are tested using the skin-detection pixel island
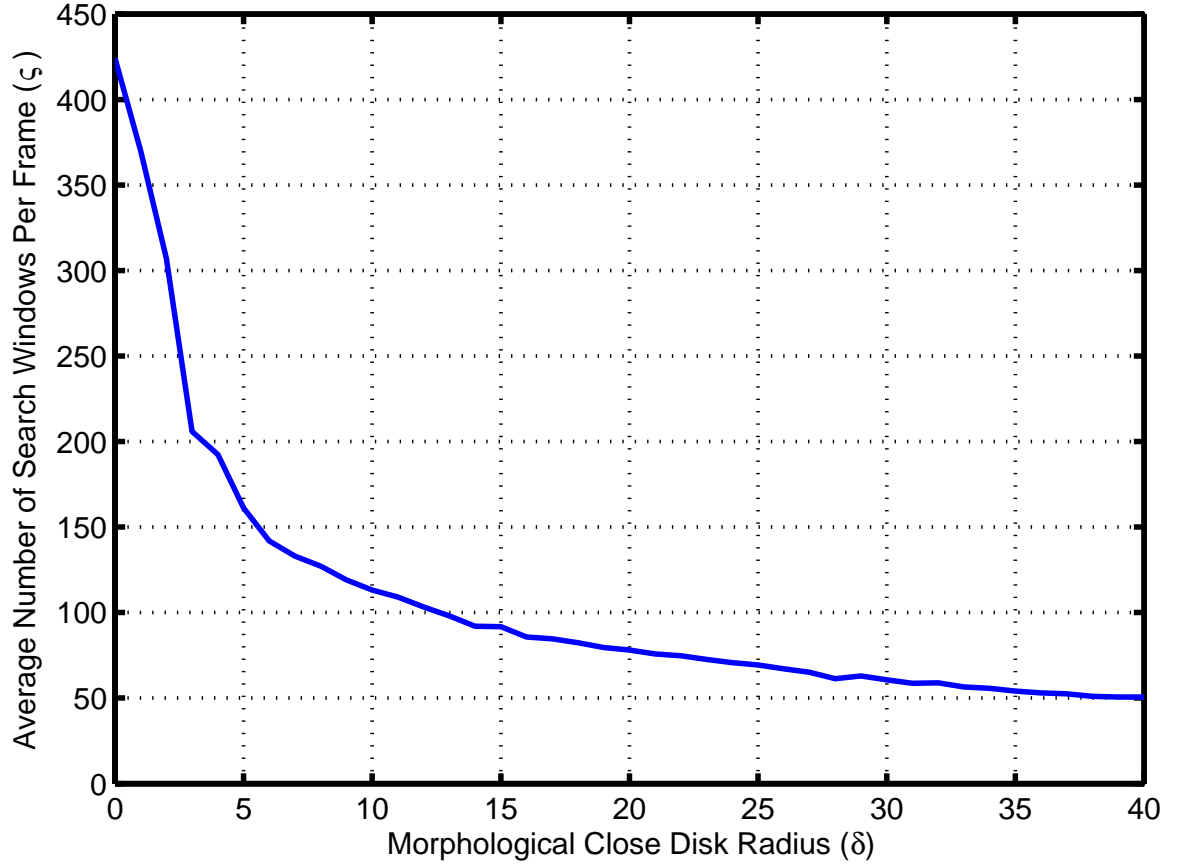
Figure 4.7:   Search windows generated as a function of morphological close disk radius (δ).

top-cuing method with $\Delta v = 15$ and slop factor $\zeta = 0$. The effect that morphological closing of skin-detection pixel islands has on HOG-based dismount detection is explored in Section 4.5.5.

Another method of reducing search windows is applying a threshold on skin-detection pixel island size (η_A) is varied from 0 to 20 pixels. Figure 4.8 illustrates how varying η_A affects the number of search windows produced for the entire HST3 data set of 42 images. For simplicity, all images are tested using the skin-detection pixel island top-cuing method with $\Delta v = 15$ and slop factor $\zeta = 0$. The effect that skin-detection pixel island thresholding has on HOG-based dismount detection is explored in Section 4.5.5.
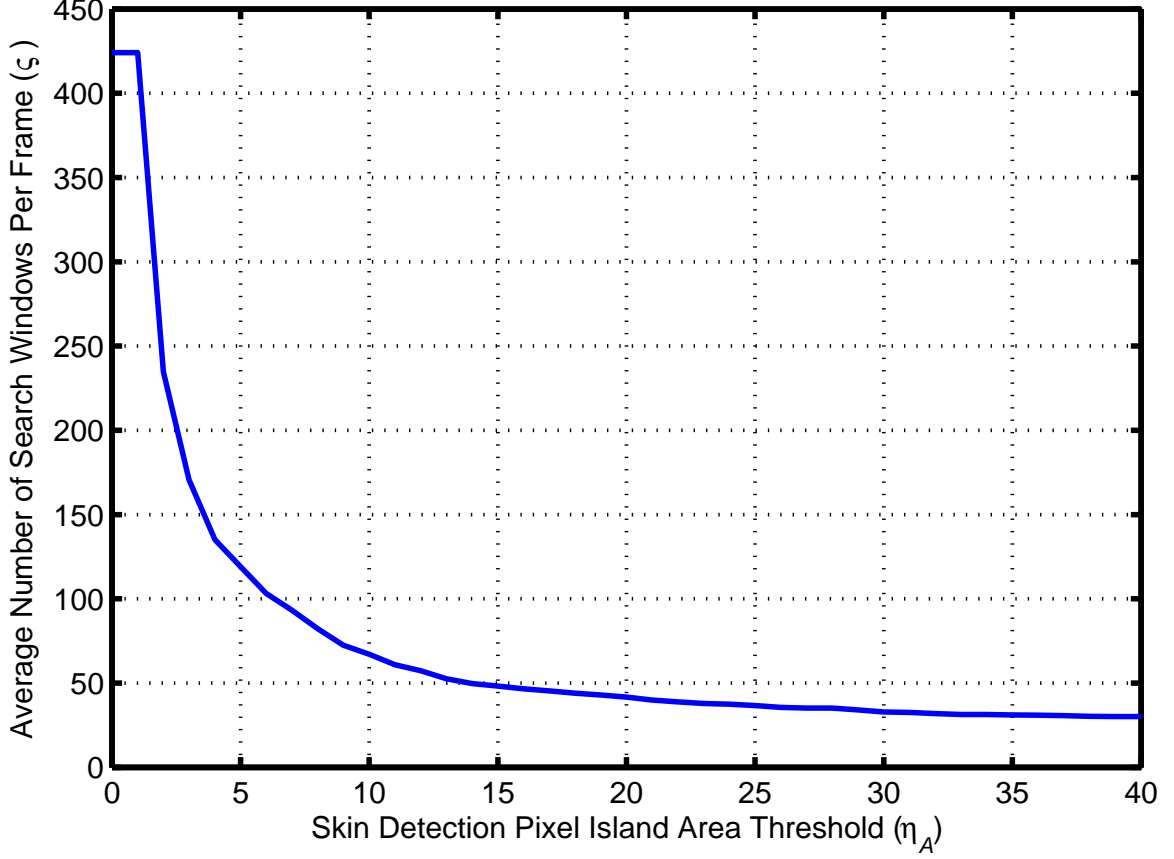
Figure 4.8:   Search windows generated as a function of threshold (η_A).

4.5   HOG-based Dismount Detector Results

4.5.1   Scoring Methodology.   To gauge detector performance, alarms are first pared down to the most-confident alarms using confidence-based non-maximum suppression as presented in Section 2.6.3. This reduced alarm set is then compared to the truth set. For each true dismount window $(t_i, i \in \mathbb{Z}[1, T]$, where T is the number of true dismounts) in the test image set, if the coverage statistic between any alarm window $(a_i, i \in \mathbb{Z}[1, A]$ where A is the number of alarms) and t_j in the same image is greater than a threshold $(\Omega(t_j, a_i) > 0.25$ as suggested by [25]), then the object in t_j is considered to have been detected and the number of detected objects (D) is incremented by one. No matter how many alarms beyond one match t_j, only one

detection is registered. The probability of detection is therefore

$$P_D = \frac{D}{T}. \tag{4.1}$$

Note that only dismounts in the scene that are upright, not partially occluded, and whose truth-window height is greater than or equal to $h_{\min} \in \theta_w(h_{\min} = 72)$ are considered in scoring. Detecting or missing people in vehicles, on bicycles, partially occluded, crouching/sitting, or shorter than $h_{\min}$ are not counted for or against the P_D calculation.

For every alarm window (a_i) in an image, if there is no true dismount window (t_j) that matches it $(\Omega(a_i, t_j) > 0.25$ as suggested in [25]) then the number of false alarms (F) is incremented by one. Therefore, the number of false positives per frame (FPPF) is

$$\text{FPPF} = \frac{F}{U}, \tag{4.2}$$

where U is the number of images tested. Note that only dismounts in the scene that are upright, not partially occluded, and whose truth-window height is greater than or equal to $h_{\min} \in \theta_w$ $(h_{\min} = 72)$ are considered in FPPF scoring. Any false alarms or rejections of people in vehicles, on bicycles, partially occluded, crouching/sitting, or shorter than $h_{\min}$ are not counted for or against the FPPF calculation. The scoring methodology presented in this section is consistent with the methodology used in [25].

4.5.2 Training the HOG-based Dismount Detector and Validation on Daimler Benchmark Imagery. In order to validate the HOG algorithm implemented in this research effort, it is important to replicate results from another recent research effort [25]. Using the Daimler Benchmark dataset provided by [25], the HOG detector is trained using 15,660 known positive dismount image patches and 15,660 randomly-selected known negative image patches. For this thesis, a MATLAB® adaption of SVM-Light [34] is utilized for training a linSVM and for making predictions after the
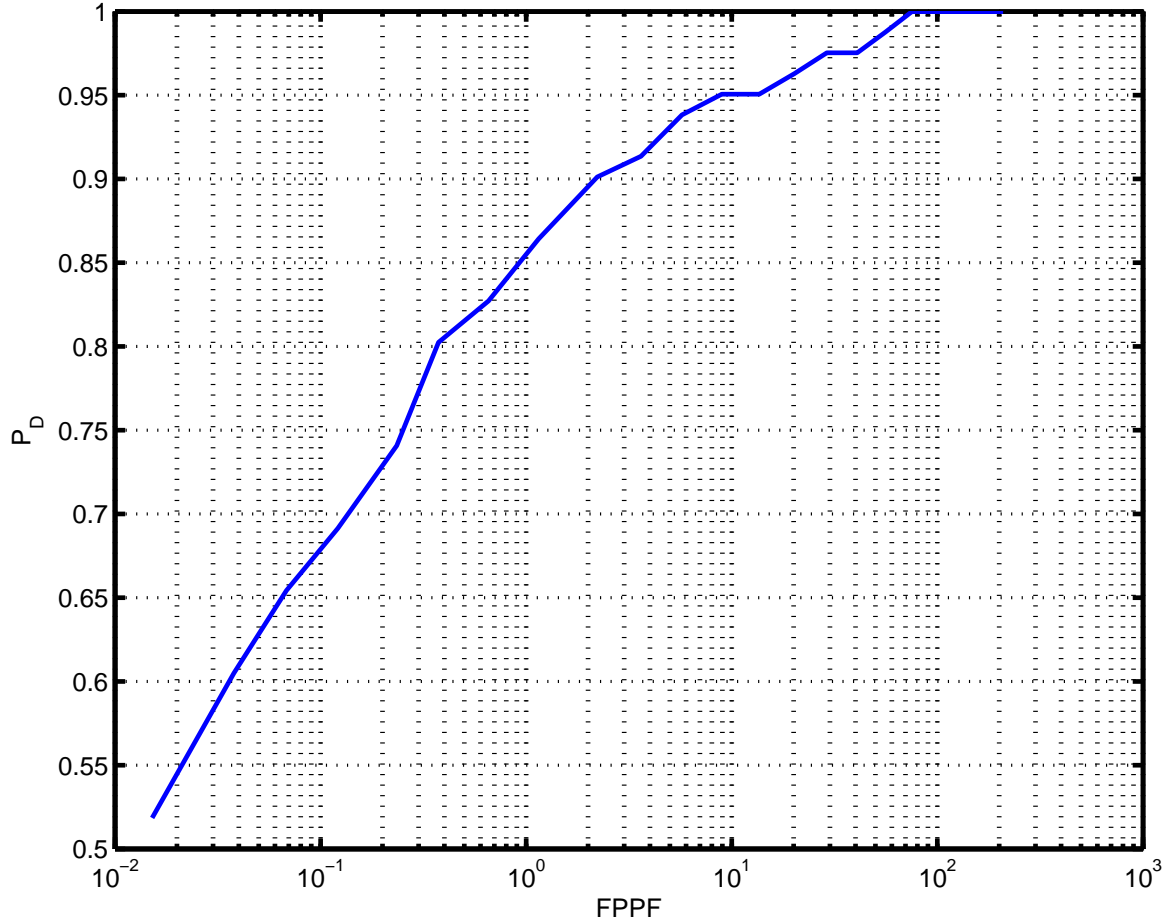
Figure 4.9: HOG-based dismount detector performance on Daimler Benchmark data using the same scoring techniques described in [25].

linSVM is trained. Due to time constraints, only one bootstrapping step is performed to enhance the detector with an additional 15,660 hard false positives[1].

Due to processing time constraints, only 264 test images out of 21,790 (approximately 1%) are tested to validate the performance of the HOG-based dismount detector implemented in this thesis effort. The images chosen for testing are a random subset of all test images that contain dismounts. The performance of the detector on this subset of test imagery is depicted in Fig. 4.9.

---

[1]The authors of [25] note that it takes several months to train the classifier with multiple bootstrapping steps. Their observation has been validated by this thesis effort, which required several weeks to train the classifier with one bootstrapping step.

Table 4.9:   HOG-based dismount detector validation results.

Reference FPPF	P_D from [25]	P_D from Fig. 4.9
10^1	≈ 0.98	≈ 0.95
10^0	≈ 0.87	≈ 0.86
10^{-1}	≈ 0.65	≈ 0.68

For validation, identical operating points are compared between the ROC curves in Fig. 6 (d), page 2189 of [25] and Fig. 4.9 of this thesis. The comparative results are listed in Table 4.9. The results depicted in Fig. 4.9 closely match the results reported in [25].

4.5.3   Truthing Methodology Considerations.   A few observations are worth noting between the borders around positive training samples and those around truth windows in the Daimler Benchmark data set. Figure 4.10 (top) illustrates five random examples of training images from the Daimler Benchmark data set. Figure 4.10 (bottom) illustrates five random examples of how the test data from the Daimler Benchmark are truthed. Note that in the training samples there is significantly more space between the bounding boxes (the edges of each image patch) and the dismount than is present in the test imagery. The borders around training samples are intentionally added to prevent edge effects from adversely affecting HOG calculations [19], [25]. Adding borders to the training samples can be viewed as an artificial bias for the detector in favor of larger scales than the truth-window scales, significantly affecting how the detector performance is scored.

It is useful to consider an "apples-to-apples" comparison in terms of alarm window versus truth window scales when scoring the dismount detector. In order to make such a comparison, either the alarm window must be rescaled to match the truth windows or vice versa. For the purposes of discussion, windows that are at *bordered-scale* are defined as windows that include borders around a dismount (i.e., similar to those in Fig. 4.10 (top)). Windows that are at *borderless-scale* are defined as windows with no borders around a dismount (i.e., similar to those in Fig. 4.10 (bottom)).

4-23

Figure 4.10:   Examples of Daimler Benchmark bounding-box differences. Training images (top row) have additional space between the dismount and the bounding box compared to test images (bottom row).

From Section 4.5.1, it is clear that the coverage statistic plays a pivotal role in how the detector is scored. Noting the bounding-box differences in Fig. 4.10, consider how they affect the coverage statistic. Assuming each positive training image patch has a 12-pixel border of background pixels around the dismount on average (as stated in [25]) and each truth window puts no such border around the same dismount, the best possible coverage statistic value between a perfectly-scaled and positioned search window (in terms of how the detector is trained) and the corresponding truth window is significantly less than the ideal $\Omega = 1$.

From visual inspection of 10 randomly-selected positive training samples (not pictured here), the 12-pixel border assumption appears to be inaccurate for the Daimler Benchmark data set. From visual inspection, there are approximately 10 pixels of background space above and below a given dismount within the training patch, while the horizontal space between the dismount and the vertical edges of the bounding boxes vary significantly as a function of dismount aspect in the image (as illustrated in Fig. 4.10 (top)).

$$\Omega(a_i, t_j) \approx 0.775$$
$$\Omega(a_k, t_j) \approx 0.775$$

$$\Omega(a_i, t'_j) = 1.000$$
$$\Omega(a_k, t'_j) \approx 0.563$$

$$\Omega(a'_i, t_j) \approx 0.838$$
$$\Omega(a'_k, t_j) \approx 0.838$$
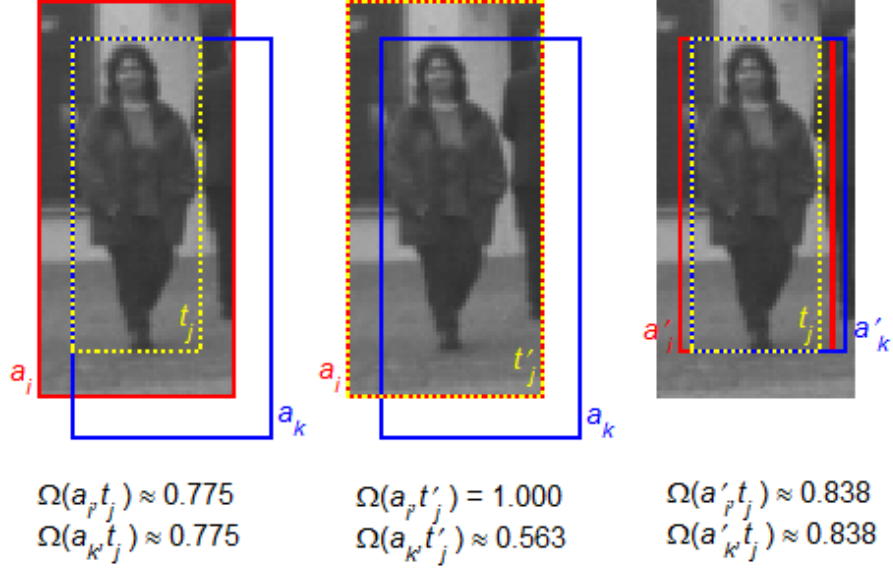
Figure 4.11: Example of best possible coverage given bounding-box differences. Red boxes indicate what the detector considers to be a "perfect detection," where a_i is at bordered-scale and a'_i is at borderless-scale. Yellow boxes indicate truth windows, where t_j is at borderless-scale and t'_j is at bordered-scale. Blue boxes indicate how minor shifts in the alarm window affect the coverage statistic, where a_k is at bordered-scale and a'_k is at borderless-scale. Dashed lines are used to aid visibility of boxes whose boundaries overlap.

Figure 4.11 (left) illustrates the coverage between the size of the window used in training (a_i at bordered-scale in red) and the truth window (t_j at borderless-scale in yellow), which is consistent with the truthing methods of the Daimler Benchmark test imagery and scoring methods used in [25].

Note that a "perfect detection" when the alarm window is at bordered-scale and the truth window is at borderless-scale corresponds with coverage $\Omega(a_i, t_j) \approx 0.775 < 1$. The truthing scheme illustrated in Fig. 4.11 (left) biases scoring in favor of alarm windows that are closer to borderless-scale. Furthermore, this truthing scheme is insensitive to minor shifts of the alarm window (illustrated by a_k in blue) in the x and y-directions, as shifting the alarm window several pixels in any direction results in the same coverage value ($\Omega(a_k, t_j) \approx 0.775$).

Figure 4.11 (middle) illustrates a new version of the truth window t'_j that is at bordered-scale. Note that a perfect detection using this truthing scheme results in coverage $\Omega(a_i, t'_j) = 1$ (therefore no bias on scale when scoring) and coverage is sensitive to all shifts of the alarm window ($\Omega(a_k, t'_j) \approx 0.563$).

To convert the truth windows to bordered-scale ($t_j \rightarrow t'_j$), truth windows should be expanded to the scale of the training sample windows (as depicted in Fig. 4.10 (top)). Each truth window is expanded by Δt_y (the equivalent of 10 pixels at the scale of the window since it is observed that a border of approximately 10 pixels exists above and below training samples at $s_n = 1$) above and below the window. An equal number of pixels (Δt_x) is added to the left and right borders of the truth window until a ratio of 2:1 (since $w_y = 96$ and $w_x = 48$) is reached. The scaled additive factor Δt_y is calculated by

$$\begin{aligned} \Delta t_y &= \frac{h \times 10}{w_y - 20}, \\ &= \frac{h}{7.6}, \end{aligned} \tag{4.3}$$

where h is the height of t_j. The scaled additive factor Δt_x is calculated by

$$\Delta t_x = \frac{h + 2\Delta t_y - 2w}{4}, \tag{4.4}$$

where w is the width of t_j.

Figure 4.11 (right) illustrates a new version of the alarm windows a'_i and a'_k that are at borderless-scale. To convert the alarm windows to borderless-scale Δt_y pixels are removed from the top and bottom of the alarm windows and Δt_x pixels are removed from the left and right of the alarm windows. Equation (4.3) and Eqn. (4.4) are still used to calculate Δt_y and Δt_x respectively, except h and w now refer to the respective height and width of the alarm window a_i or a_k. Note from Fig. 4.11 (right) that when alarm windows are converted to borderless-scale, truth windows are no longer forced to have the same height-to-width ratio as alarm windows. Variance in

truth window widths therefore can lead to variations in the perceived performance when scoring occurs since it may be impossible to achieve perfect overlap of the alarm and truth windows. Furthermore, there are similar issues with multiple alarm window positions yielding the highest possible coverage value as with the original truth method (Fig. 4.11 (left)), though the highest possible coverage value is significantly higher.

Figure 4.12 demonstrates how ROC curves for the Daimler Benchmark imagery are affected by all three truthing methodologies discussed. The blue curve depicts the ROC curve calculated using the techniques described in [25] (i.e., truth windows at borderless-scale and alarm windows at bordered-scale). The red curve depicts the resulting ROC curve when truth windows are converted to bordered-scale (i.e., both alarm and truth windows are at bordered-scale). The green curve depicts the resulting ROC curve when alarm windows are converted to borderless-scale (i.e., both alarm and truth windows are at borderless-scale). Note that the underlying detector does not change, but how the detector is scored (and therefore the ROC curve) *does* change. The differences appear to be minor in Fig. 4.12 (differences of no more than 0.05 P_D at the same FPPF), but it is important to note that there is literally no change in how the detector operates or the data on which it operates. Differences on this scale may be acceptable when considering different random subsets of a data pool, but not when testing on the same identical data set with the same underlying detector. The purpose of this discussion is to highlight the importance of specificity when reporting how a sliding-window detector performs.

For all further scoring, the truth windows are converted to bordered-scale using the technique described above. This is done for several reasons:

1. The full range of the coverage statistic is utilized.

2. There is only one alarm scale and position that results in a perfect score ($\Omega = 1$).

3. Scoring is an "apples-to-apples" comparison (i.e., the truth windows and alarm windows are all at bordered-scale).

4. Alarm windows and truth windows have the same aspect ratio.

4-27
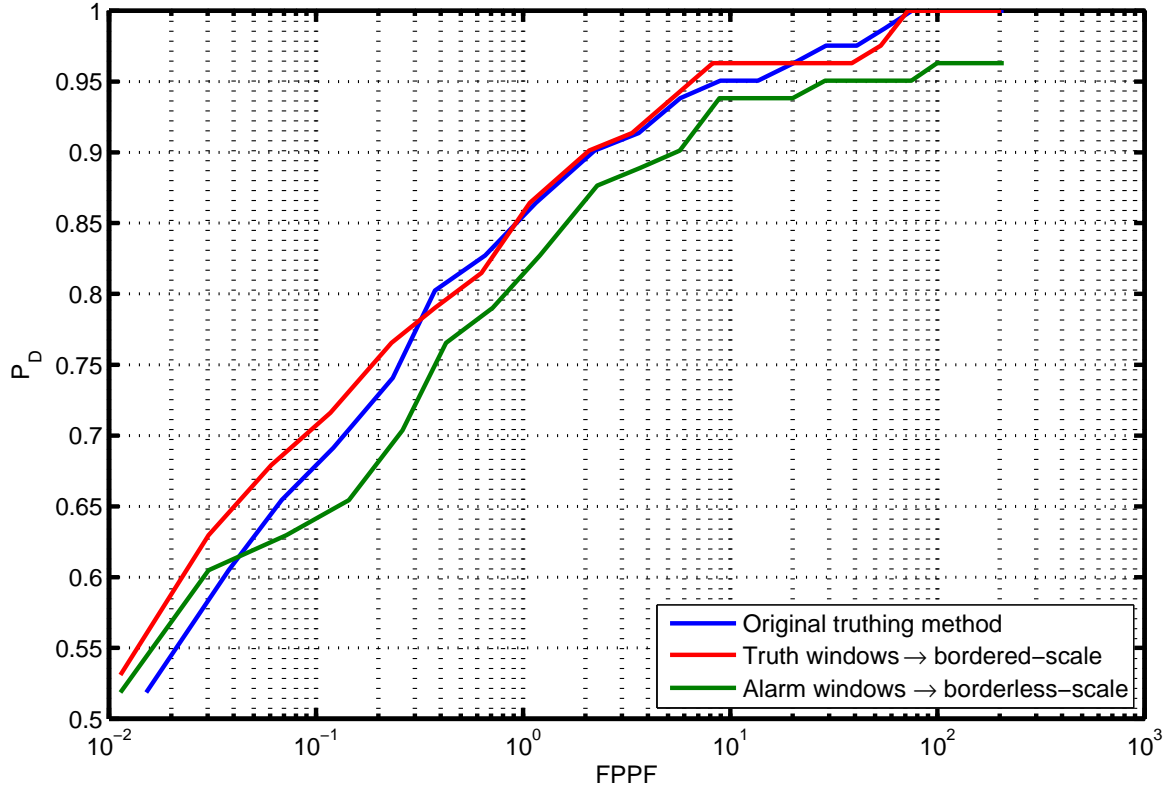
Figure 4.12:   Comparison of how truthing techniques affect the HOG-based dismount detector performance on Daimler Benchmark data. The blue curve results from using the original truth windows. The red curve results from expanding the truth windows to bordered-scale (as described in Eqn. (4.3) and Eqn. (4.4)). The green curve results from shrinking the alarm windows to borderless-scale (as described in Eqn. (4.3) and Eqn. (4.4)).

4.5.4   Full Image Search Results for HST3 Imagery.   In [25], only dismounts that are greater than or equal to 72 pixels in height are scored when the ROC curves are generated. This is intuitive for fair scoring since the search window scale is limited to be no less than 72 pixels in height ($h_{\min} \in \theta_w$ where $h_{\min} = 72$).

As a baseline for comparison with the methodology proposed by this thesis effort, the HOG-based dismount detector trained on the Daimler Benchmark training set is applied to the HST3 data set using the exact same methods and parameters used in the validation comparison in Section 4.5.2. The results of this baseline full search of the HST3 imagery are depicted in Fig. 4.13. The red curve denotes results
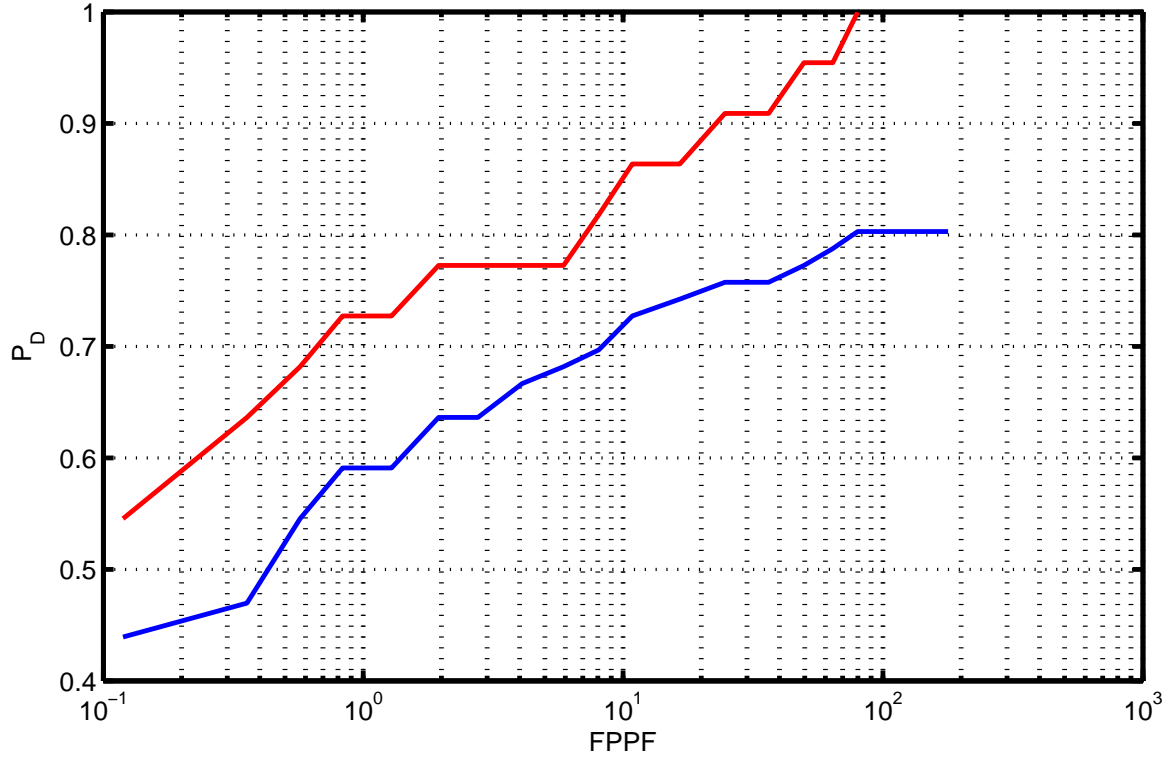
Figure 4.13:   Full search results for HST3 data. The red curve represents scoring of upright dismounts that are not occluded and whose height $h \geq 72$ pixels. The blue curve represents scoring of upright dismounts that are not occluded regardless of height.

from scoring only targets that are greater than or equal to 72 pixels in height. The blue curve denotes results from scoring all dismounts that are in an upright position and not partially-occluded.

Note that the stair-stepping in the P_D dimension occurs for both curves in Fig. 4.13. This stair-stepping is a result of the small number of dismounts that meet the 72-pixel height requirement in the dataset (66 dismounts with no restriction on height, 22 dismounts with $h \geq h_{\min}$, where $h_{\min} = 72$ pixels).

4.5.5   Skin-detection-cued Search Results for HST3 imagery.   There are several key parameters that can be adjusted to affect skin-detection-cued dismount detection performance. These include Δu (pixel island centroid-based) or Δv (pixel island top-based) for search window positioning, δ (morphological close disk radius) or η_A

(threshold on the number of pixels in a skin detection pixel island) for reducing the number of skin-detection pixel islands, and ζ for adding shifted windows. For testing purposes, the standard sliding window parameter set (θ_w) described in Section 2.6.1 remains constant.

Assuming that all parameters above are independent of one another (i.e., no synergistic effects between parameters), finding the best parameter set involves setting all parameters to a constant value except for the parameter under test. As the best parameters are found from each test, they are subsequently used to help determine the best values for other parameters. For the purposes of this thesis, "best parameters values" are experimentally determined by sweeping values of the parameter and visually comparing resultant ROC curves. This subjective approach is a form of greedy search and therefore has no guarantee of optimality.

The skin detection algorithm used for this parameter study is the rules-based detector with parameters $b_1 = -1$, $b_2 = -0.02$, $c_1 = 0.26$, $c_2 = 0.93$. The rules-based detector is chosen for computational efficiency and parameter adjustability.

Power thresholding on estimated reflectance values is used prior to NDSI and NDGRI calculations because there are many deeply-shadowed areas in each HST3 image tested in this thesis effort. In those shadowed areas, all estimated reflectance values are below 0.02, which is near the HST3 noise floor. Values near the noise floor have wildly varying NDSI or NDGRI values across the entire range of $\mathbb{R}[-1, 1]$ since sensor noise is dominating the original pixel values. Therefore, pixels that are less than 0.02 estimated reflectance at 1080nm are set to a very small constant (to prevent divide-by-zero errors) at all wavelengths. This forces all NDSI values for those pixels to be 0, guaranteeing they will be ignored by the skin detector.

For all parameter trade-off studies in Sections 4.5.5.1-4.5.5.3, only limited ROC curves are generated to determine relative performance of different parameter values. The full range of performance is not explored due to the computation time necessary

to process each ROC curve when a high number of false alarms are present and the large number or ROC curves necessary to make valid assessments.

4.5.5.1 Δu Versus Δv Trade-off Results. Likely the most critical parameter is the offset value used to position primary search windows relative to skin-detection pixel islands. Two methods of determining the y-location for the search window are presented in Section 3.3.4. First, the best subjective values for Δu and Δv are determined. To experimentally determine the best subjective value for Δu–the scaled distance from the top of a search window to the centroid of a skin-detection pixel island–control values $\zeta = 0$ (i.e., no additional shifted search windows are generated), $\delta = 0$, and $\eta_A = 0$ (i.e., no modifications are made to skin-detection pixel islands) are used.

To roughly determine the range of values needed for Δu, Δu is first varied from 5 to 40 in increments of 5 and linSVM predictions are made. To generate comparative ROC curves, the threshold on prediction values (η_τ) is varied from -2 to 10 in increments of 0.2. The ROC curves for each Δu-value are visually compared and the parameter value associated with the dominant curve is chosen ($\Delta u = 15$). Next, the range for Δu is varied from 12 to 20 by increments of 2 and ROC curves are generated for comparison using the method mentioned above. The best value from this test is $\Delta u = 16$.

Finally, the range for Δu is varied from 14 to 18 by increments of 1 and ROC curves are generated for comparison using the method mentioned above. The resulting ROC curves are depicted in Fig. 4.14. Note that there is no clear winner evident in Fig. 4.14. The red curve almost always dominates the blue, teal, and purple curves (except for a few cross-over points), but the green and red curves battle for dominance all along the range of performance. Since the red curve dominates at low FPPF, and performance between the red and green curves crosses over frequently at high FPPF, it is determined that the red curve is the "winner" in this subjective comparison, therefore $\Delta u = 16$ (red) is selected for use in future testing.
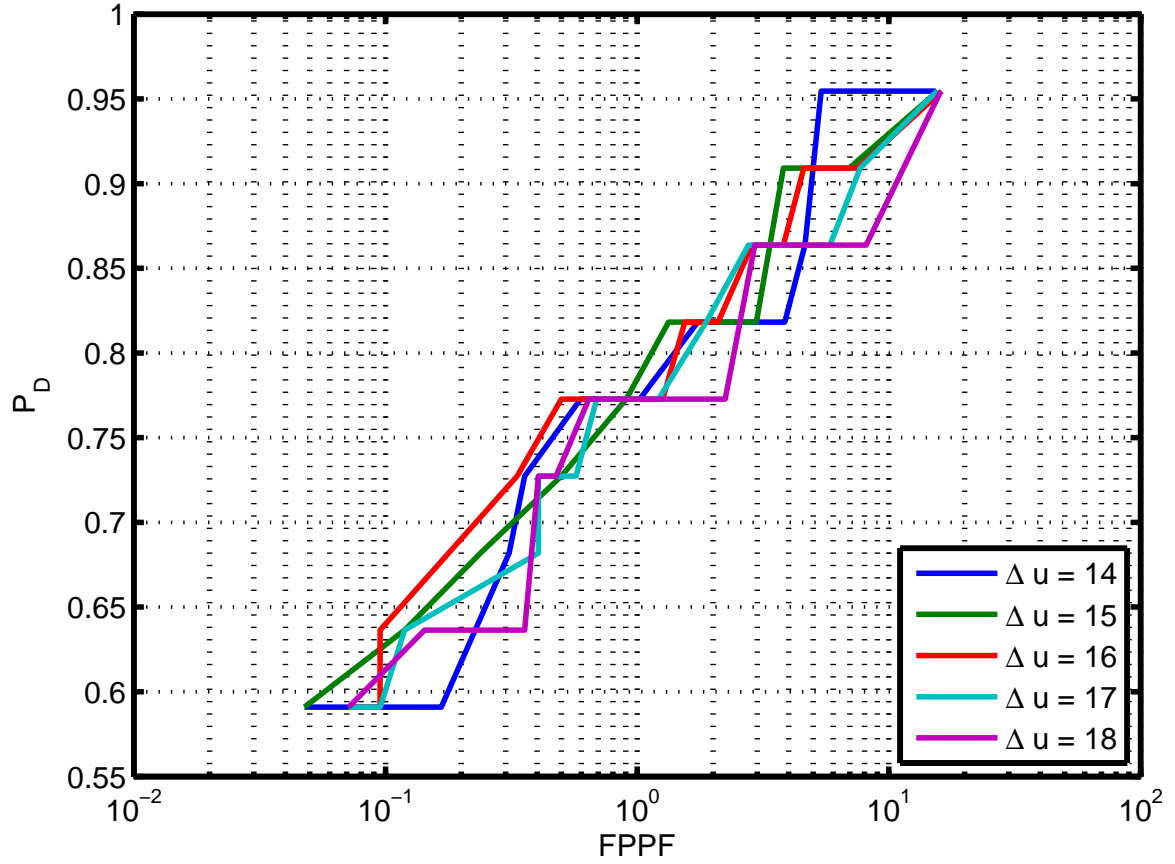
Figure 4.14:   Performance comparison for multiple centroid-cueing parameter (Δu) values.

To experimentally determine a reasonable value for Δv, the same control variable values are used as those for the Δu assessment. Similar coarse-to-fine sweeps of Δv values are used to generate ROC curves for comparison. The resulting ROC curves from the finest sweep of Δv values are depicted in Fig. 4.15. As with the Δu comparison above, no ROC curve associated with a Δv value clearly dominates in Fig. 4.15. The blue curve, while marginally dominant at higher FPPF, is grossly dominated at lower FPPF. However, the teal curve dominates (or is almost tied with the dominant curve) more often than it is dominated by other curves, therefore it is determined that $\Delta v = 15$ (teal) is a reasonable value to use.

To determine whether the Δu or Δv method performs better, ROC curves for $\Delta u = 16$ and $\Delta v = 15$ are compared in Fig. 4.16. From this comparison, it is
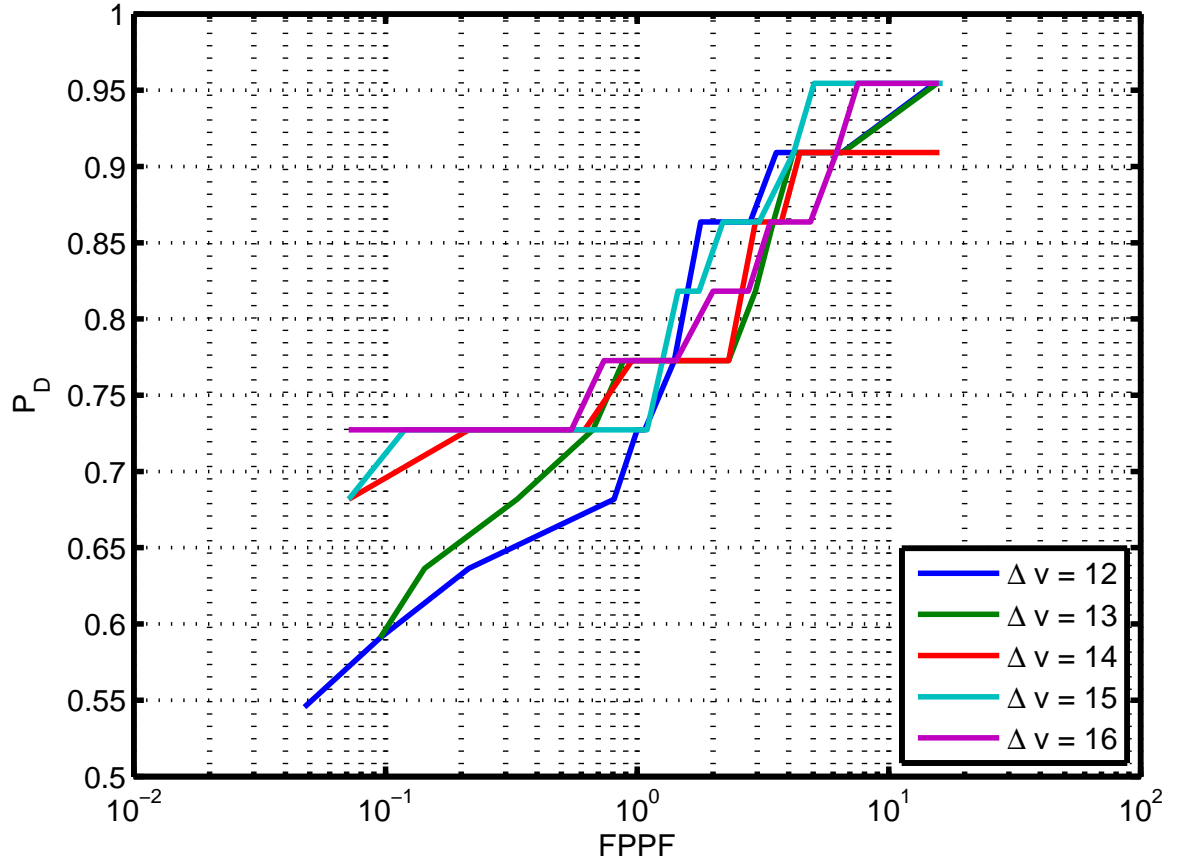
Figure 4.15:   Performance comparison for multiple top-cueing parameter (Δv) values.

determined that the $\Delta v = 15$ (red) method has a marginal performance advantage over the $\Delta u = 16$ (blue) method on the limited data set used in this research effort. Therefore, $\Delta v = 15$ is used for all further testing. However, the results are not definitively conclusive, especially since variations in clothing and hairline are minimal in the data set tested, so this choice may not be globally suitable beyond the scope of this data set.

4.5.5.2   Morphological Close Disk Radius Versus Area Threshold Trade-off Results.   Next, reasonable values for the morphological close disk radius (δ) and the threshold on skin detection pixel island area (η_A) for reducing the number of skin detection pixel islands are experimentally determined. To determine a reasonable
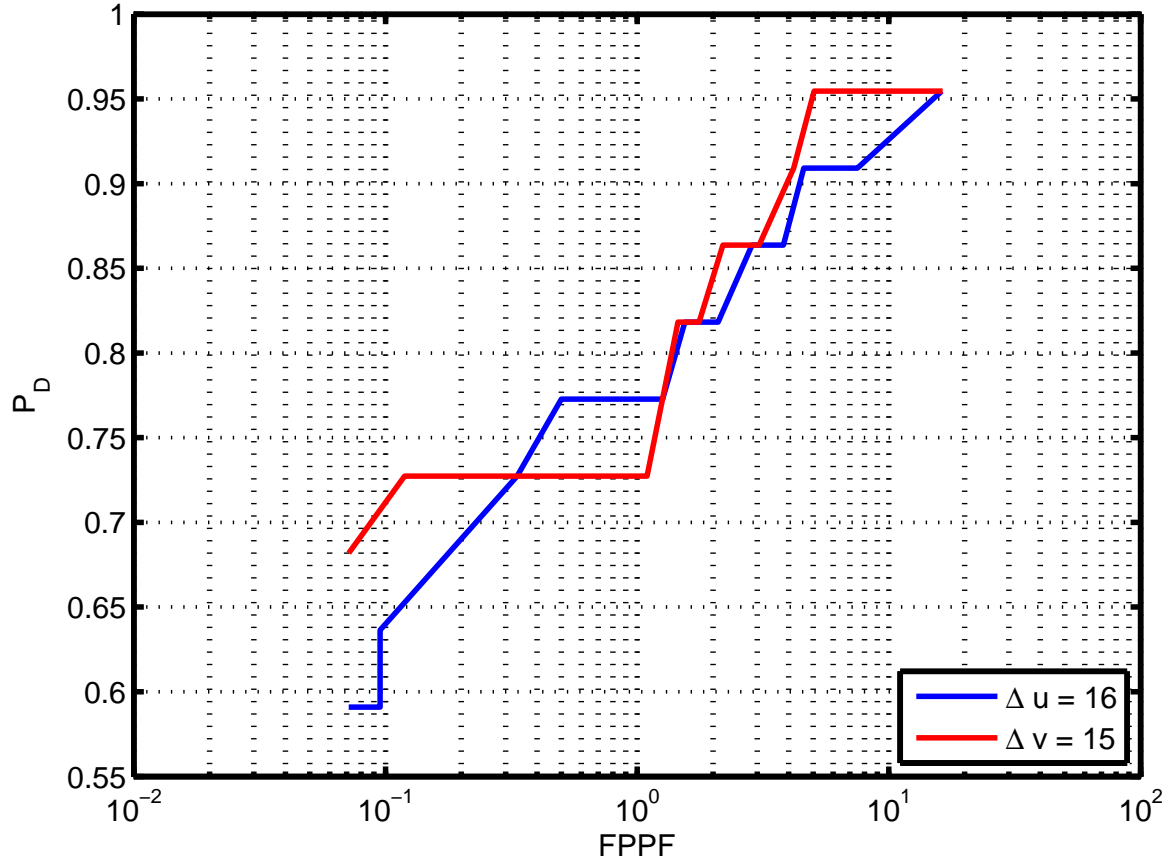
Figure 4.16:   Performance comparison of Δu and Δv cueing methods.

value for δ, control values $\Delta v = 15$ and $\zeta = 0$ (no additional shifted search windows are generated) are used.

To experimentally determine a reasonable value for δ, coarse-to-fine sweeps of δ values (again similar to the Δu comparison methodology above) are used to generate ROC curves for comparison. The resulting ROC curves from the finest sweep of δ values are depicted in Fig. 4.17. Again, there is no clearly-dominant ROC curve, but the red and teal curves approach dominance. Since the teal curve dominates the red curve in two regions while the red curve only dominates the teal curve in one region, it is determined that $\delta = 8$ (teal) is a reasonable value to use.

To experimentally determine a reasonable value for η_A, the same control variable values are used as those for the δ assessment. Similar coarse-to-fine sweeps of η_A values are used to generate ROC curves for comparison. The resulting ROC curves from the
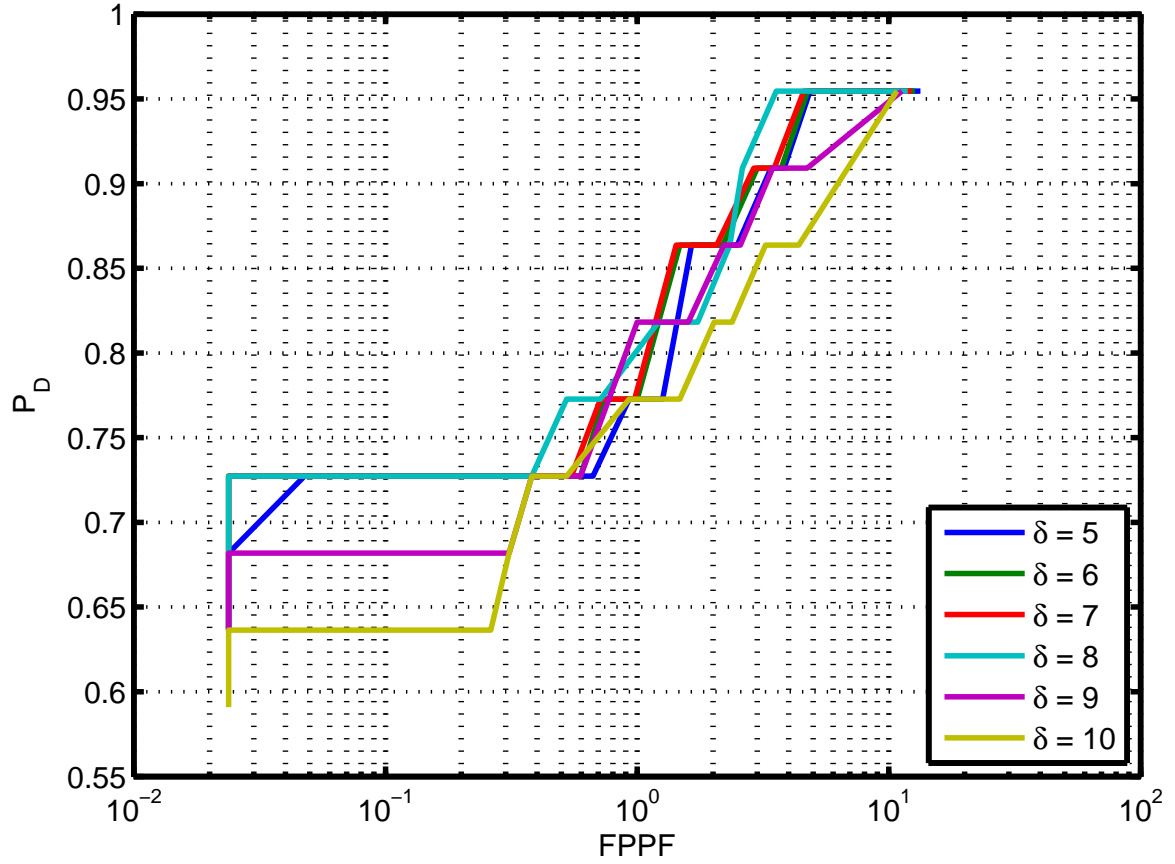
Figure 4.17:   Performance comparison for multiple morphological close disk radius (δ) values.

finest sweep of η_A values are depicted in Fig. 4.18. Since the results for all values of η_A from 38 to 42 are identical, $\eta_A = 38$ is chosen as a reasonable value to use because the least amount of information is destroyed.

To determine whether the δ or η_A method performs better, ROC curves for $\delta = 8$ and $\eta_A = 38$ are compared in Fig. 4.19. From this comparison, it is determined that the $\eta_A = 38$ method has a significant performance advantage over the $\delta = 8$ method on the limited data set used in this research effort. Therefore, $\eta_A = 38$ is used for all further testing.

Note that the $\eta_A = 38$ method may make it impossible to detect distant dismounts or dismounts with very small areas of exposed skin. The choice of whether
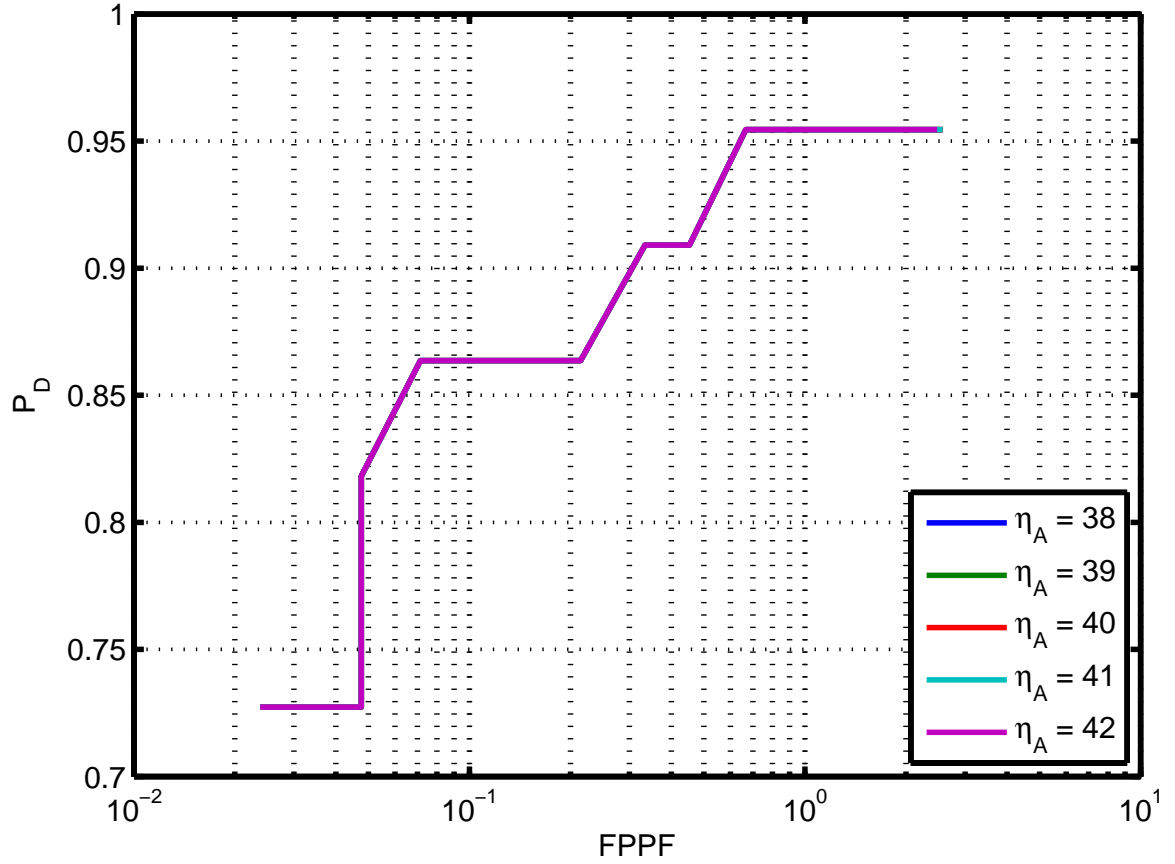
Figure 4.18:   Performance comparison for multiple skin detection pixel island area threshold (η_A) values. Note that all curves are identical. Therefore, only the curve for $\eta_A = 42$ appears to be present because all other curves lie directly beneath it.

to use η_A thresholding or δ-radius disk close operations to suppress spurious skin detection pixel islands should be considered based on the operational environment.

4.5.5.3 ζ Trade-off Results.   Next, a reasonable number of shifted search windows in each direction (ζ) to add to the base skin-detection-cued search windows is experimentally determined. To determine a reasonable value for ζ, control values $\Delta v = 15$, $\delta = 0$ (i.e., no morphological close operations), and $\eta_A = 38$ are used while $\zeta \in \mathbb{Z}[0, 3]$.

Figure 4.20 depicts ROC curves for each ζ-value. The performances of all ROC curves in Fig. 4.20 where $\zeta > 0$ are very similar. The ROC curve for $\zeta = 0$ has better
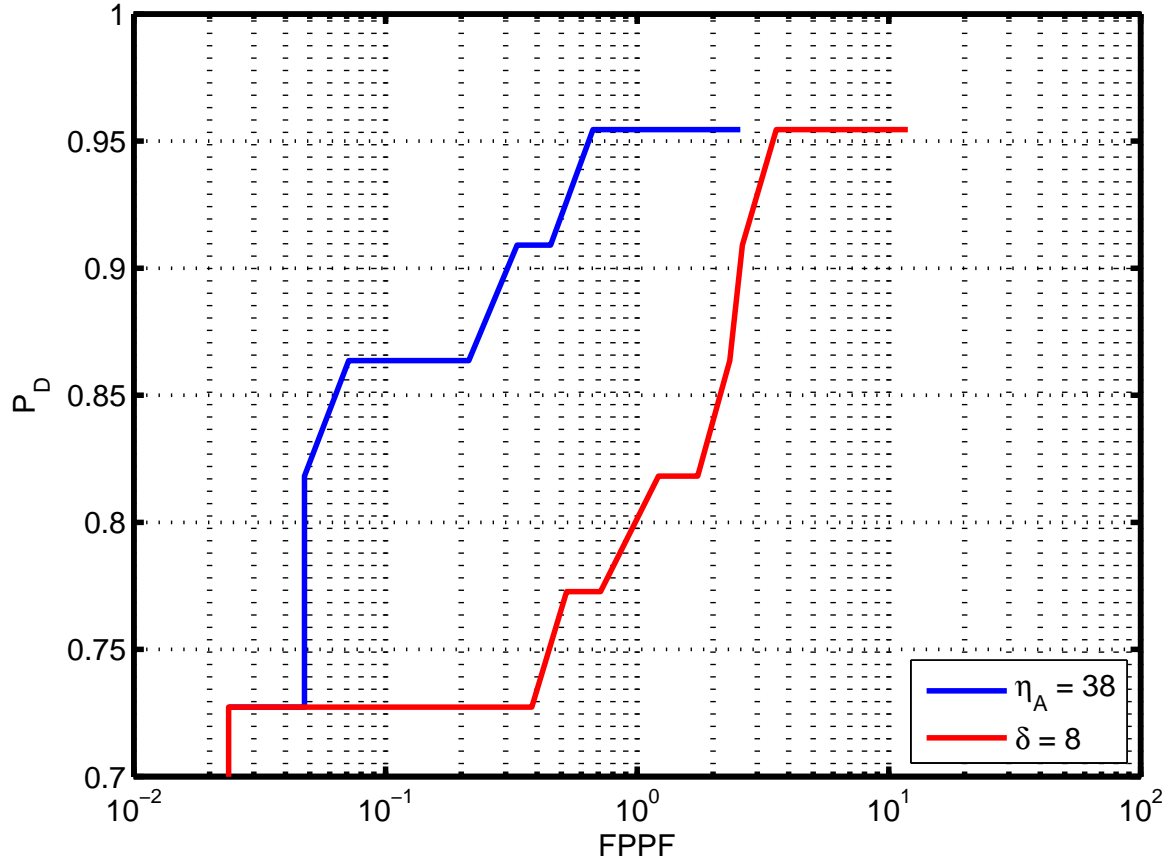
Figure 4.19:   Performance comparison of δ and η_A methods.

false alarm performance in general. Based on the results depicted in Fig. 4.20, $\zeta = 0$ (blue) is experimentally determined to be a reasonable value to use since the blue curve marginally dominates performance across most of the performance range.

4.5.6   Full Search Versus Skin-detection-cued Search Performance Results for HST3 Data.   Figure 4.21 depicts the comparative ROC curves for the full search HOG-based dismount detector on the HST3 data versus the skin-detection-cued HOG-based dismount detector using the experimentally determined parameter values ($\Delta v = 15$, $\eta_A = 38$, and $\zeta = 0$). At 95% probability of detection, the skin-detection-cued HOG-based dismount detector outperforms the full-search method in terms of false alarm suppression by an order of magnitude in false positives per frame. Additionally, the ROC curve for the skin-detection-cued HOG-based dismount detector dominates
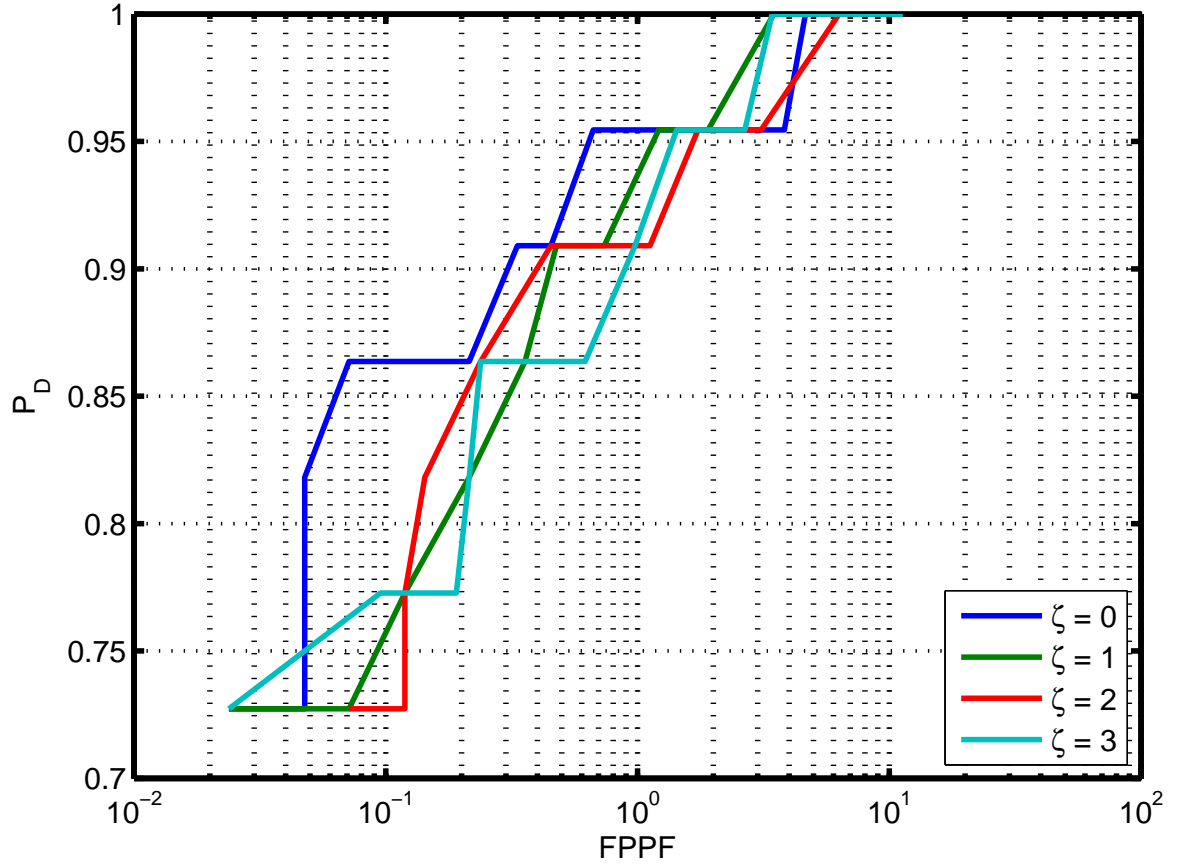
Figure 4.20:   Performance comparison for multiple slop parameter (ζ) values.

the ROC curve for the full-search HOG-based dismount detector across the entire range of operating points. This indicates that the skin-detection-cued HOG-based dismount detector significantly outperforms the full-search HOG-based dismount detector for the data set tested.

The total number of search windows generated for each image in the HST3 data set are depicted in Fig. 4.22. Using skin-detection-cueing to generate search windows with $\zeta = 0$ leads to a reduction of the search space by nearly three orders of magnitude for the HST3 data, depending on the number of dismounts in the scene.

4.6   Summary

This chapter begins by describing the data sets used in this research, followed by exploration of multiple aspects of skin detection including features for false-alarm
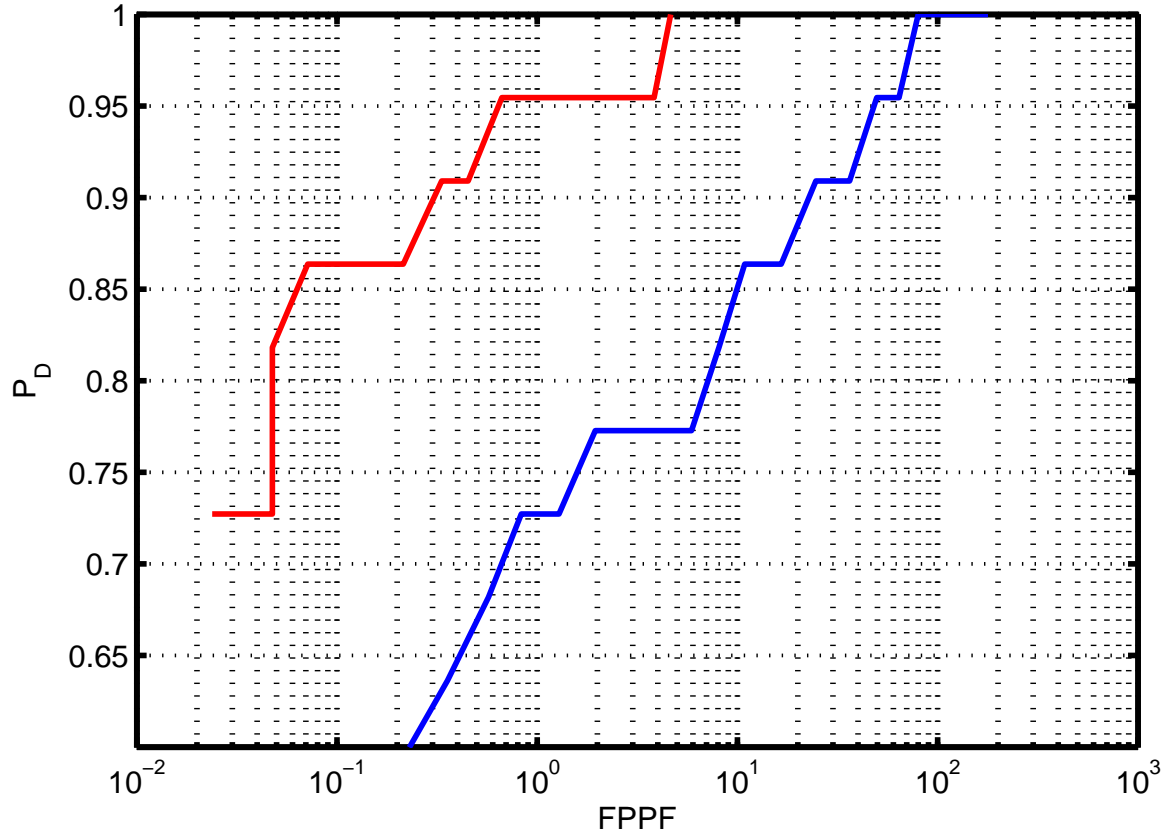
Figure 4.21: Full search (blue) versus skin-detection-cued search (red) performance for HST3 data.

suppression and skin detection algorithms. It is concluded that the NDGRI feature is better for suppressing false alarms during the skin detection process with the data tested. It is also concluded that the rules-based and LRT-based skin detection algorithms perform almost identically on the data tested, thus making it logical to use the rules-based skin detector for further testing due to computational efficiency.

Next, search window generation is explored noting how image resolution, the number of skin detection pixel islands, and the slop factor affect the number of search windows generated. It is concluded that methods for intelligently reducing the number of skin detection pixel islands can significantly reduce the number of search windows generated.

The performance of the baseline dismount detector is validated next by reproducing the methods outlined in [25] on the same data set they used. It is con-
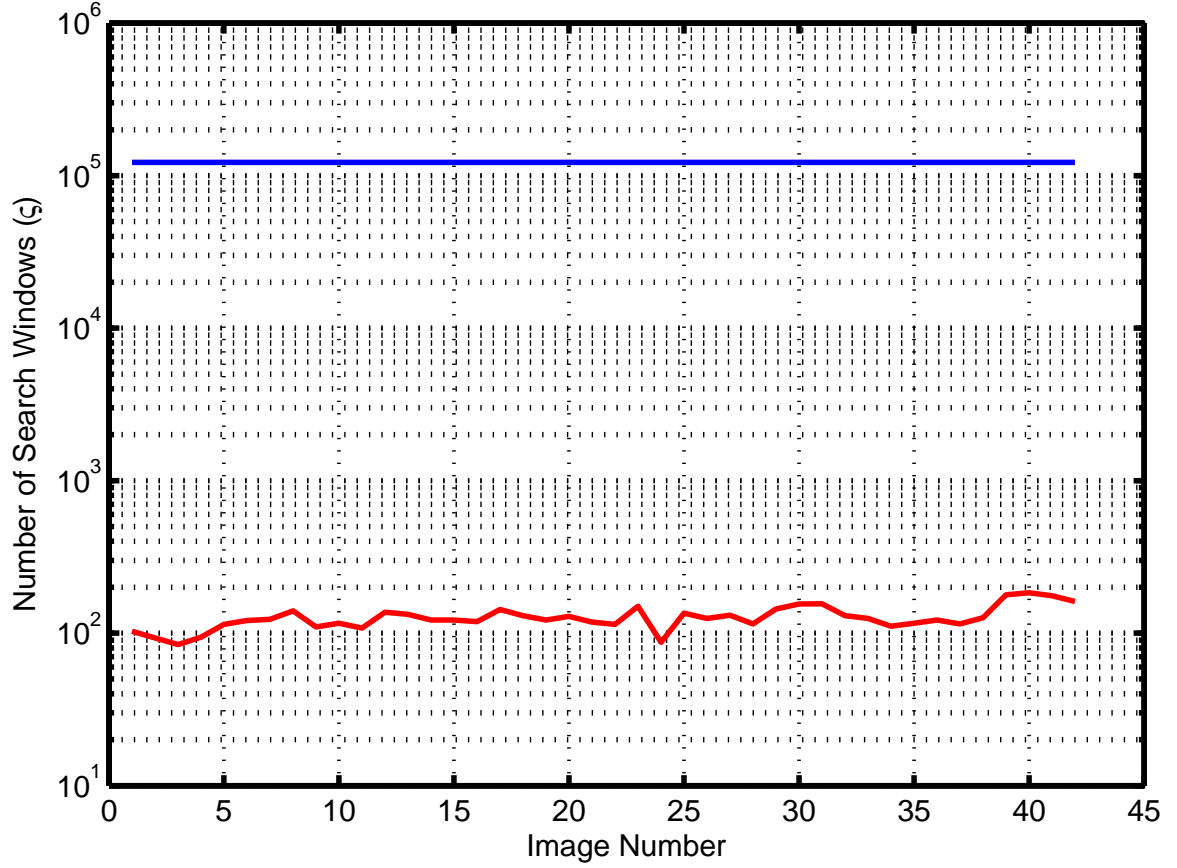
Figure 4.22:   Total number of search windows generated for full search (blue) versus skin-detection-cued search (red) using HST3 data.

cluded that the methods used in this thesis produce the same results as those used in [25]. A discussion on truthing techniques concludes that minor differences in scoring methodology produce measurable difference is performance curves. Therefore, scoring methodology should be explicitly described when presenting results for a sliding-window detector.

Next, search window positioning parameter sets are experimentally determined. It is concluded that reasonable parameters to use for generating skin-detection-cued search windows for the data set used in this research are $\Delta v = 15$, $\delta = 0$, $\eta_A = 38$, and $\zeta = 0$.

Finally, a comparison is made between the performance of the baseline full-search dismount detector and the skin-detection-cued dismount detector. It is con-

cluded that the skin-detection-cued dismount detector requires nearly 3 orders of magnitude less search windows for the data set tested. Furthermore, the skin-detection-cued dismount detector produces nearly 2 orders of magnitude less false positives per frame than the full-search method at 0.95 probability of detection.

V. Conclusions and Future Work

This chapter summarizes the work accomplished in this thesis effort and provides recommendations for future work. First, a summary of the methods and conclusions is provided, followed by recommendations for future work. Finally, contributions made by this thesis effort to the sensor modeling, skin detection, and dismount detection research communities are provided.

5.1 Summary of Methods and Conclusions

The primary focus of this thesis is to employ skin detections to cue a dismount detector based on histograms of oriented gradients (HOG). For skin detection, a trade-off study is conducted coupling the normalized difference skin index (NDSI) feature for skin detection with the normalized difference vegetation index (NDVI) feature or normalized difference green-red index (NDGRI) feature for false alarm suppression. It is concluded that the NDGRI feature provides better false alarm suppression overall than the NDVI feature.

Next, a trade-off study is conducted comparing the performance of a rules-based skin detector and a likelihood-ratio test (LRT)-based skin detector (developed in this thesis effort) on both modeled and imaged hyperspectral data. In order to develop the LRT-based skin detector, this thesis effort develops methodology for simulating the response of an arbitrary sensor by applying sensor noise parameters to laboratory-measured spectral data. While the LRT-based skin detector performs slightly better than the rules-based skin detector in general, the performance differences between the two detectors is not significant. Therefore, since the rules-based skin detector is significantly less complex than the LRT-based skin detector, it is concluded that the rules-based skin detector should be used in situations where detector flexibility and low computational complexity are desired.

Next, a HOG-based dismount detector is trained using training samples from the Daimler Benchmark data set provided by [25] and validated on a subset of test images from the Daimler Benchmark data set. The validation performance is almost identical

(differences of $\pm 0.03 P_D$ at the same false positives per frame (FPPF) operating points) to the results presented in [25].

A study of truthing methodology for dismounts in imagery is conducted to determine the effect of detector scale bias on scoring methodology. It is concluded that adjusting truth windows to match the scale bias introduced when training the detector gives the most accurate assessment of the detector "as trained." Using truth windows where dismounts completely fill the windows (i.e., with no space between the edges of a truth window and the dismount it contains) gives an unbiased assessment of the true performance of the dismount detector.

Next, the same full-search methodology used to validate the results in [25] is used on the HyperSpecTIR version 3 (HST3) data set. A trade-off study is then conducted to experimentally determine parameters to use when generating search windows from skin detection pixel islands. For the HST3 data set used in this thesis, it is concluded that the best experimentally-determined values to use are top-cueing with $\Delta v = 15$, thresholding of skin detection pixel islands by area with $\eta_A = 38$, and no additional "slop" windows ($\zeta = 0$). Finally, comparisons are made between the full-search and skin-detection-cued search methods in terms of performance and search space size. This skin-detection-cueing technique reduces the required search space by nearly three orders of magnitude depending on the number of dismounts in the scene, while improving the false alarm rate from approximately 50 to 0.65 false positives per frame at 95% probability of dismount detection, nearly two orders of magnitude improvement in false alarm suppression.

5.2 Recommendations for future work

There are many avenues for expansion upon this thesis effort in future work. First, significant effort should be placed on collecting a larger, more diverse database of hyperspectral or multispectral imagery. At the time of this research effort, no publicly-available high-resolution hyperspectral image database exists as a benchmark for future testing. Once significantly more data are available, the HOG-based

dismount detector should be retrained on a subset of those data so that the detector is trained on sensor-specific data. Operationally, the dismount detector should be trained on example imagery from the sensor that is employed, thus making the detector as robust as possible.

With the diversity of spectral information available in a larger hyperspectral or multispectral dataset, the HOG-based detector should be extended beyond panchromatic imagery. In [19], it is suggested that if RGB imagery are available, using the image channel with the greatest gradient magnitude for each pixel when assigning histogram votes can significantly improve performance. This technique is logical because it takes advantage of the channel containing the most contrast for edge-orientation binning. Applying this technique to hyperspectral or multispectral imagery may produce similar improvements in dismount detection performance.

Integrating skin detection cueing of a HOG-based dismount detector into a tracking framework is another natural extension of this work. Utilizing a real-time, multispectral sensor, such as the one designed in [60], will provide additional utility over line-scanning hyperspectral systems like the HST3 due to reduced operator complexity and increased frame rate.

While skin detection has been demonstrated as useful for cueing a HOG-based dismount detection system by this thesis effort, it is clearly not without limitations. The most significant limitation is that the methods developed in this thesis require exposed skin in the head/face region of a dismount. Augmenting the skin detection cueing approach with clothing detection cueing can have multiple benefits. First, if no exposed skin is available on a particular dismount, clothing may provide a reasonable cueing source. Furthermore, having additional information about clothing may improve the tracker's ability to disambiguate targets of interest.

One of the most challenging and time-consuming tasks required in the course of this thesis effort is determining which image pixels are truly skin. This task is critical for accurately gauging skin detection performance. Due to mixed-pixel effects

5-3

and human subjectivity, this task is very difficult to accomplish in any reasonable amount of time. Systems such as Digital Imaging and Remote Sensing Image Generation (DIRSIG) [2] use first-principles approaches to accurately simulate any sensor's response to a simulated scene. Since the entire scene is software-generated, perfect pixel truth is known. Incorporating the first-principles human skin model [51], [55] into a system such as DIRSIG would be beneficial not just for pixel-truthing, but also for generating a large and arbitrarily diverse data set that fits any sensor modality that can be simulated by the software. Additionally, the first-principles model of human skin should be extended to a full Bidirectional Reflectance Distribution Function (BRDF) model to incorporate angular dependencies as discussed in Section 2.7.1.

The first steps toward first-principles integration into software simulation of humans have already been taken. In [54], a 3-dimensional model of a human face is successfully populated with skin-model-generated reflectance spectra to generate part of a holistic human avatar. Adding clothing, hair, and fingernail spectra to this avatar model would complete the software simulation.

To rapidly add diversity of poses to the human avatar simulation, human motion capture systems [3] can be used to animate the avatar, a technique commonly used for assisting computer animation in theatrical movies. This would allow videos of complex motion to be generated for any arbitrary sensor modality within the spectral range of the models used to populate avatar pixel spectra. Applications of such simulation capabilities are far-reaching throughout the human measurement and signature intelligence (H-MASINT) community.

5.3 Contributions

This thesis effort makes several significant contributions to the skin detection, sensor modeling, and dismount detection research domains. In the skin detection domain, this thesis effort improves detection performance by determining the best set of several spectral features (NDSI, NDVI, NDGRI) required to improve separability of the skin class from materials outside the skin class. Additionally, multiple skin detec-

tion algorithms are compared including the LRT, which incorporates an optimality criterion.

In the sensor modeling domain, this thesis provides methodology for applying sensor noise and specular reflection components to modeled or laboratory-measured data. This allows the performance of any arbitrary imager sensitive in the spectral range of the model or laboratory-measured data to be simulated to be simulated as long as the noise components of the imager and the target geometry and BRDF are known. This is useful for evaluating sensor design prior to prototyping if the noise components of the constituent components can be approximated.

In the dismount detection domain, utilizing skin detection for cueing a HOG-based dismount detector reduces the search space required by nearly 3 orders of magnitude. Additionally, dismount detector false alarm performance is improved by nearly 2 orders of magnitude at 95% probability of detection when compared to the original full-search system. The skin-detection-cued HOG-based dismount detector developed in this thesis has the potential to make a significant contribution to the United States Air Force (USAF) intelligence, surveillance, and reconnaissance (ISR) and human H-MASINT missions.

Appendix A. Bilinear Interpolation

Bilinear interpolation is used to approximate the value at an arbitrary point within a two-dimensional set of known data. Bilinear interpolation is a combination of 3 linear interpolations (2 in the x-direction and one in the y-direction). The four points with known values ($Q_{ij}; i, j \in \{1, 2\}$ at position (x_i, y_j)) that are nearest the desired value (Z at position (x, y)) are used for the interpolation calculations (as depicted in Fig. A.1).
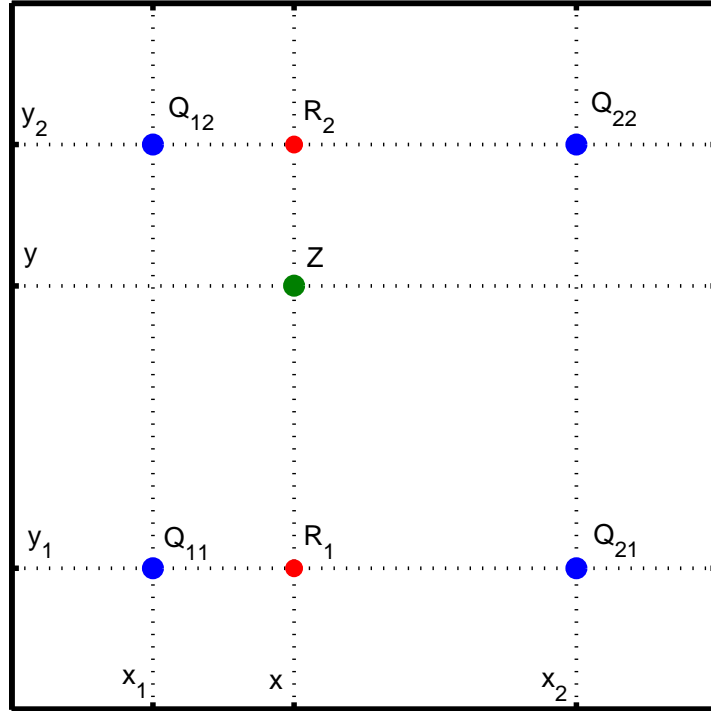

Figure A.1:   Bilinear interpolation example.

First, linear interpolation is performed to determine the intermediate values at R_1 and R_2 by

$$R_1 = \frac{x_2 - x}{x_2 - x_1} Q_{11} + \frac{x - x_1}{x_2 - x_1} Q_{21}, \tag{A.1}$$

$$R_2 = \frac{x_2 - x}{x_2 - x_1} Q_{12} + \frac{x - x_1}{x_2 - x_1} Q_{22}, \tag{A.2}$$

where x_1 and x_2 are the x coordinates associated with Q_{ij}. Finally, linear interpolation is performed to determine the value of at the desired point by

$$Z = \frac{y_2 - y}{y_2 - y_1} R_1 + \frac{y - y_1}{y_2 - y_1} R_2. \qquad (A.3)$$

Appendix B. Skin Detection Masks For All HST3 Images Used

This appendix presents the skin detection and dismount detection results for all HST3 images used in this thesis. The top window of each figure depicts the original RGB image from the HST3 imager for reference. The second image is the skin detection mask using the rules-based skin detection algorithm with $\gamma \in \mathbb{R}[0.26, 0.93]$ and $\beta \in \mathbb{R}[-1, -0.02]$. The third image depicts the dismount detection results at the 95% probability of detection operating point. The fourth image depicts the dismount detection results at the 0.05 FPPF operating point. For the third and fourth images, white boxes indicate dismount alarms that are considered hits, while red boxes indicate dismount alarms that are considered false alarms. The parameters used for cueing the dismount detector are $\Delta v = 15$, $\eta_A = 38$, and $\zeta = 0$.

(a) Original HST3 image
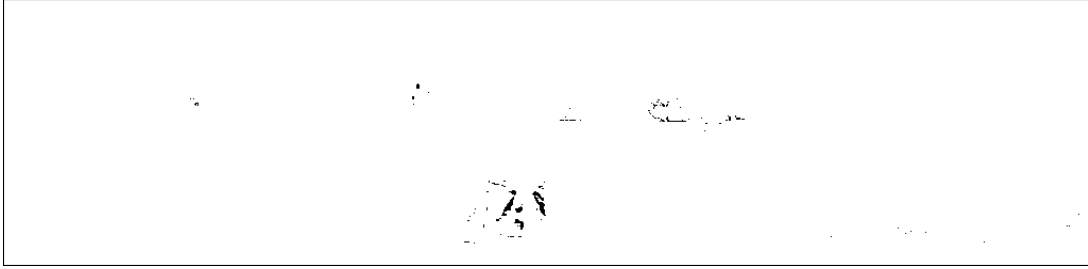
(b) Skin detection mask

(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.1:   Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 1. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

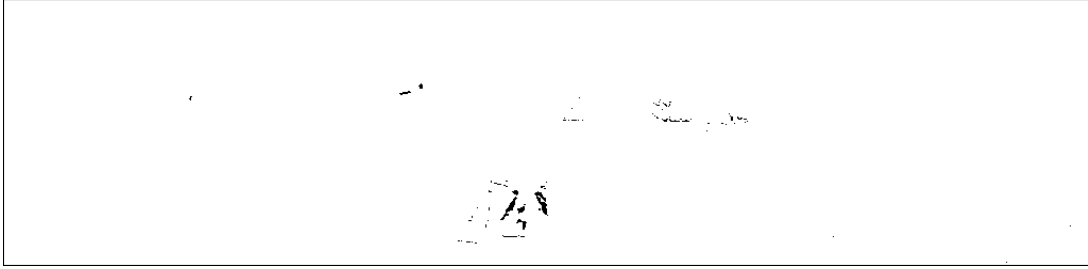(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.2:   Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 2. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

(c) Dismount detection boxes at 0.95 P_D operating point
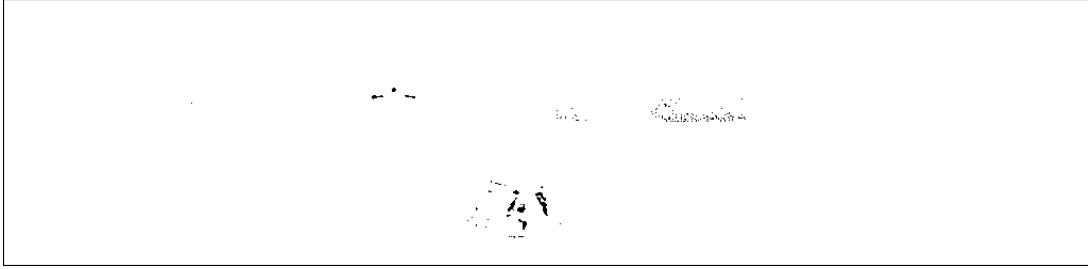

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.3: Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 3. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

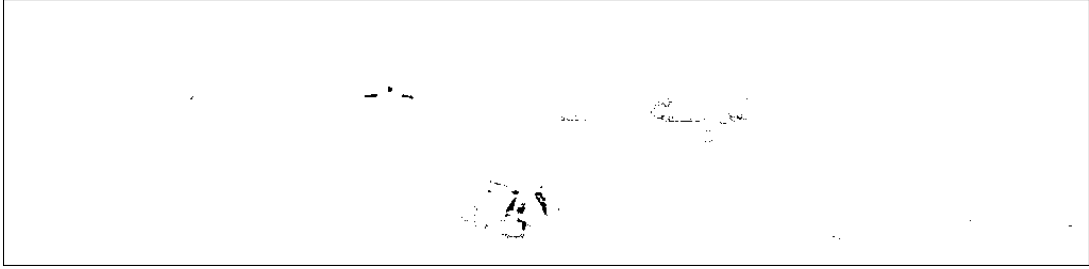(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.4: Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 4. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

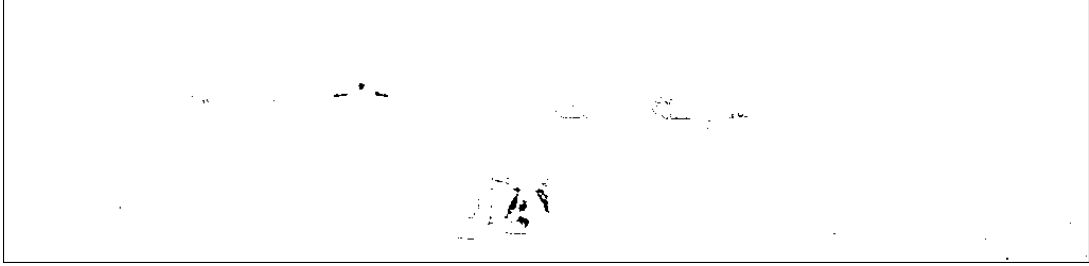(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.5:   Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 5. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

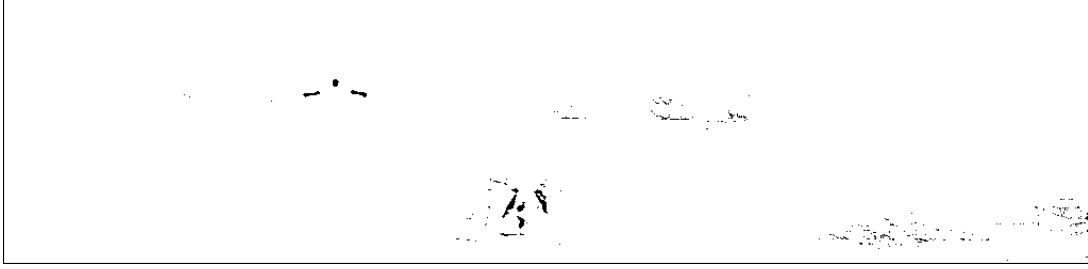(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.6:   Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 6. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

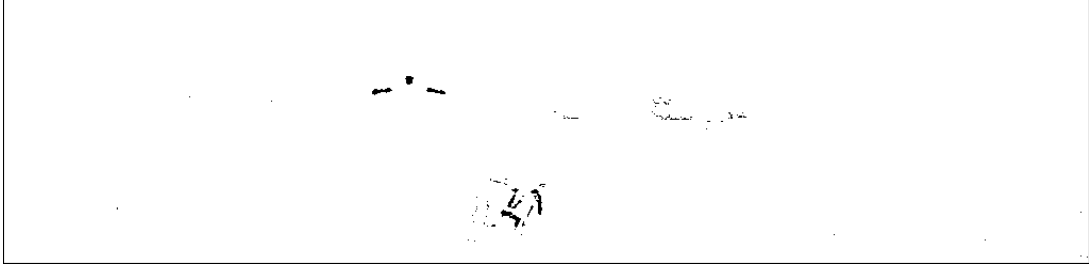(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.7: Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 7. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

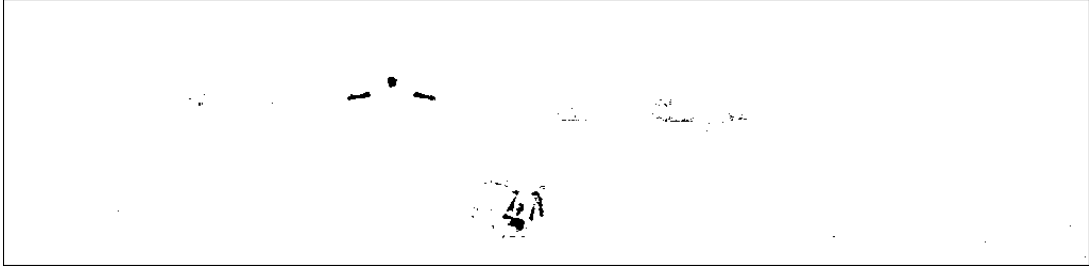(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.8: Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 8. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.9: Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 9. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

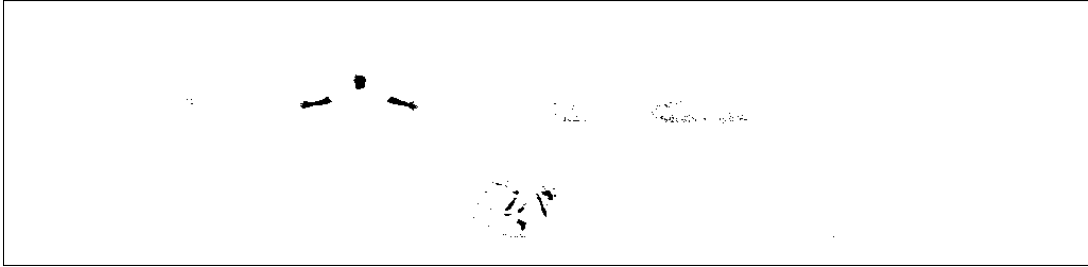(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.10:   Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 10. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

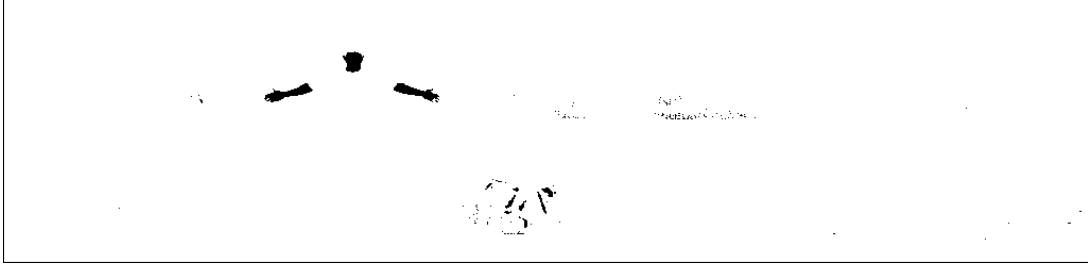(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.11: Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 11. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.12:   Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 12. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.13: Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 13. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

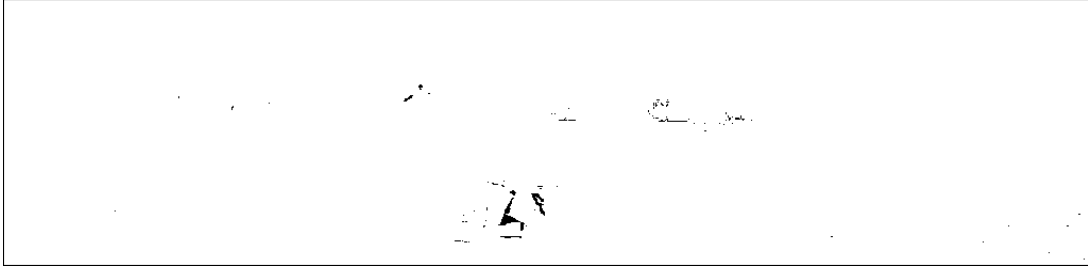(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.14:   Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 14. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

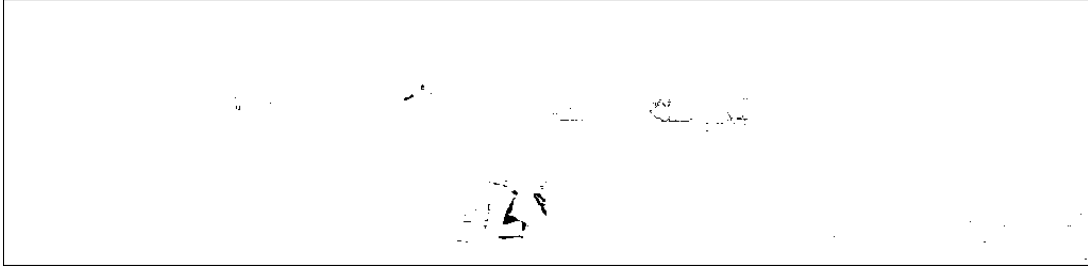(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.15:   Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 15. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

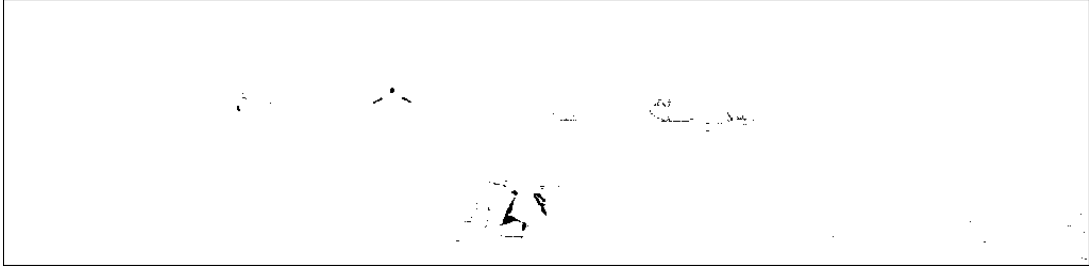(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.16: Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 16. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

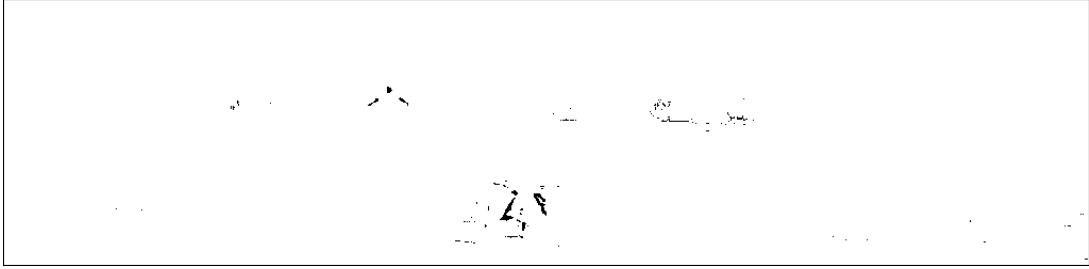(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.17:   Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 17. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

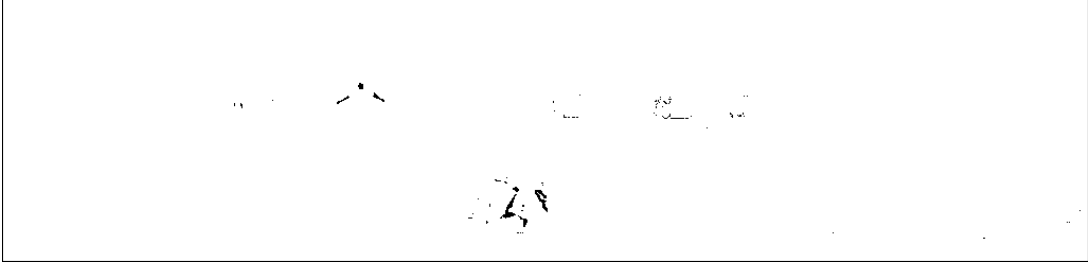(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.18:   Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 18. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

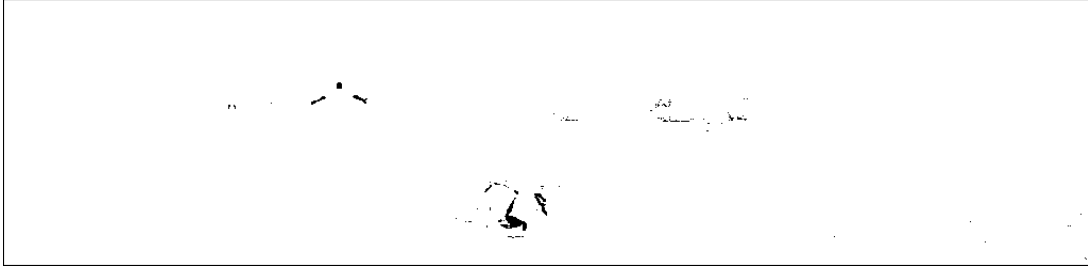(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.19: Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 19. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

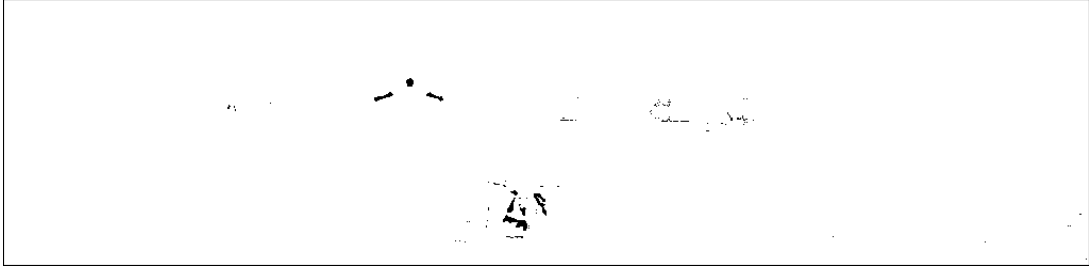(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.20:   Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 20. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point
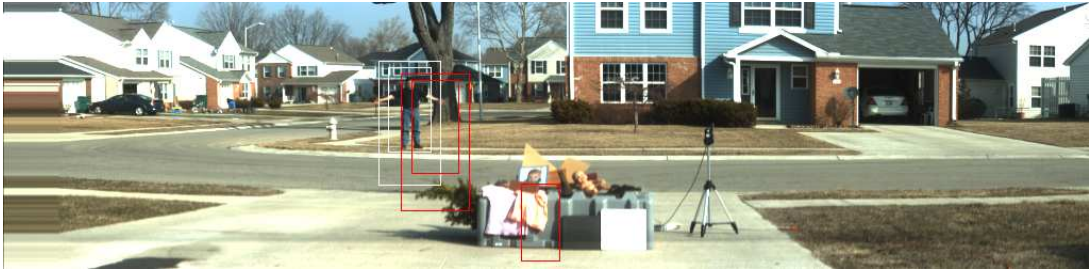
Figure B.21:   Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 21. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

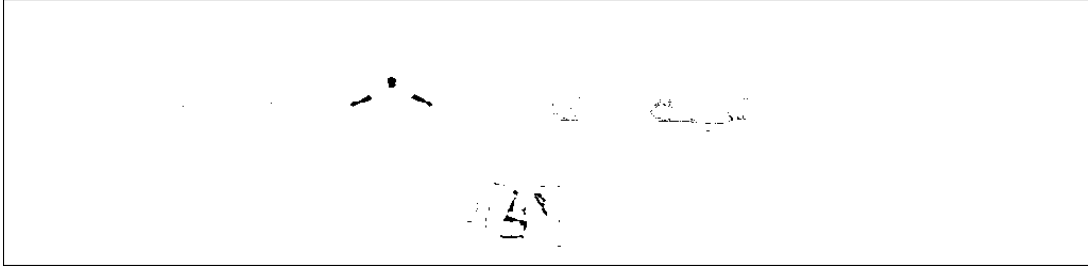(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.22: Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 22. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.
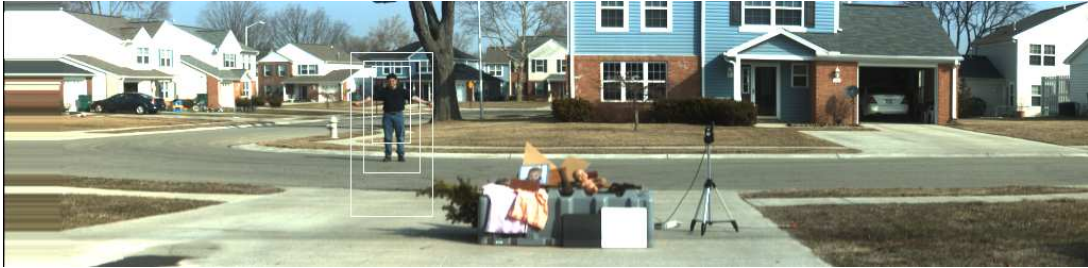
(a) Original HST3 image

(b) Skin detection mask

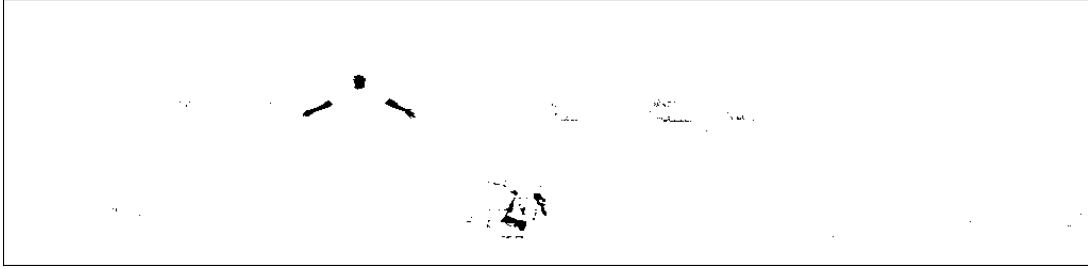(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.23: Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 23. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

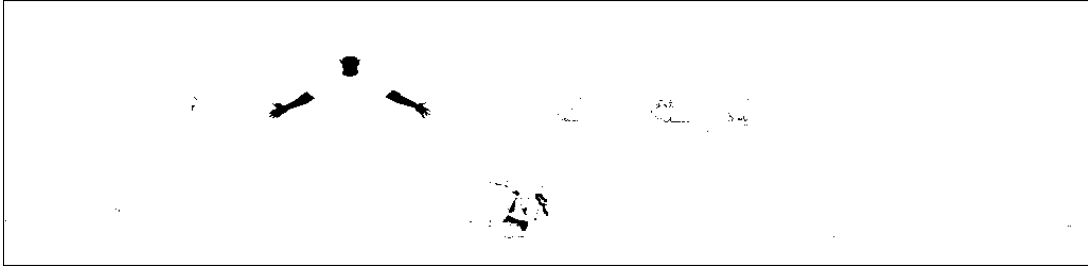(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.24:   Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 24. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

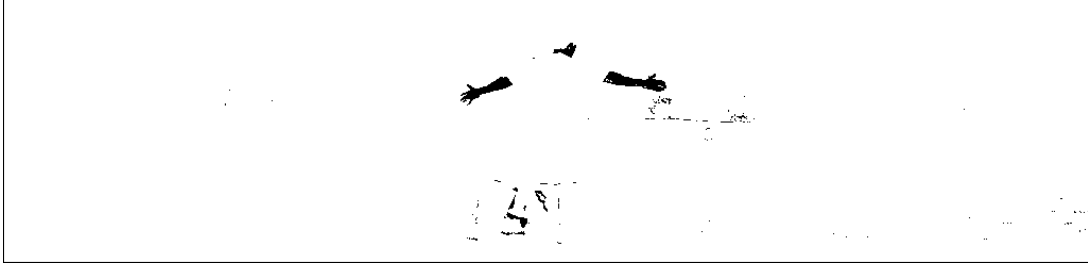(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.25: Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 25. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point
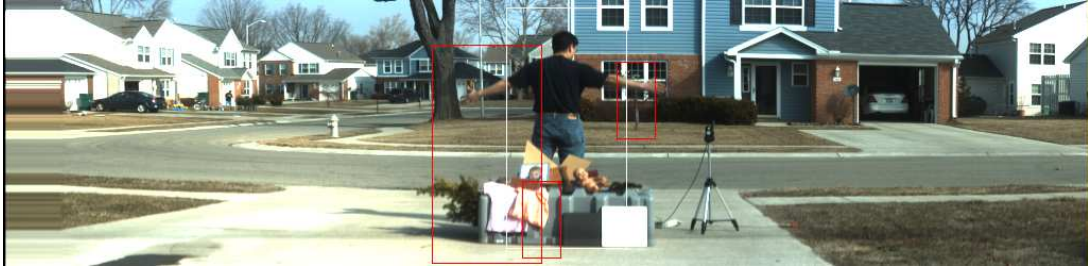
Figure B.26:   Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 26. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.
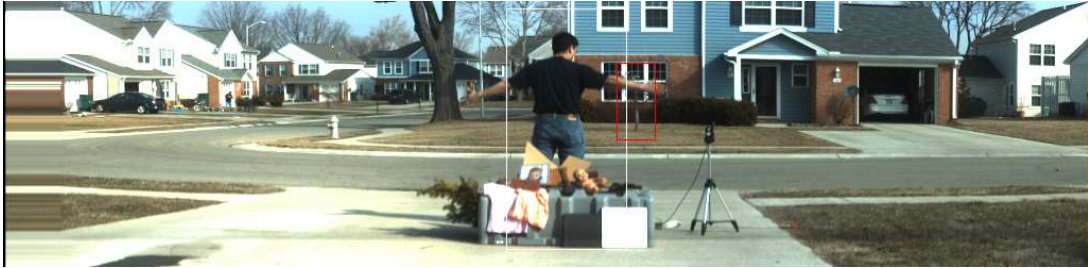
(a) Original HST3 image

(b) Skin detection mask

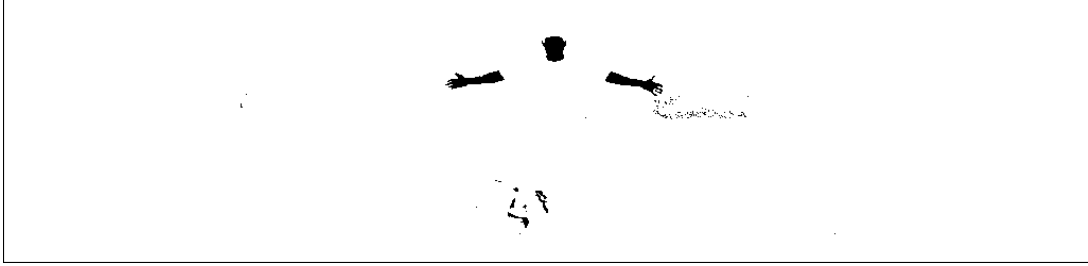(c) Dismount detection boxes at 0.95 P_D operating point

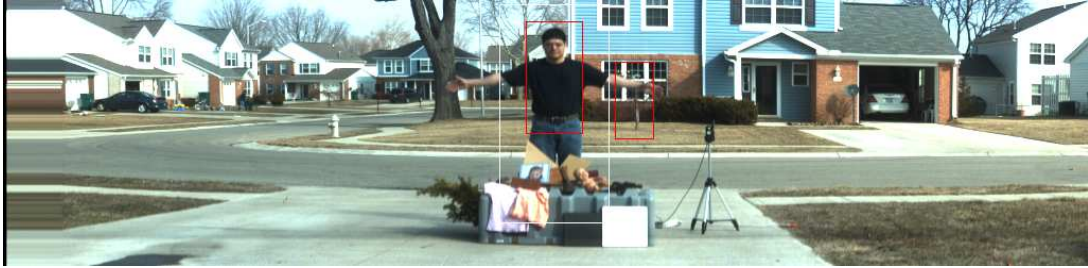(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.27: Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 27. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.28: Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 28. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.
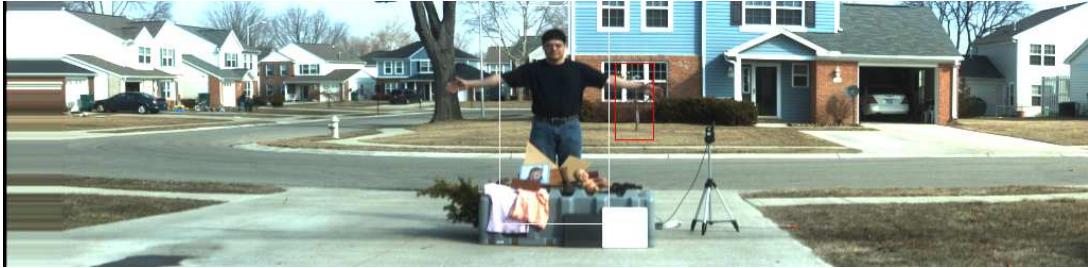
(a) Original HST3 image

(b) Skin detection mask

(c) Dismount detection boxes at 0.95 P_D operating point

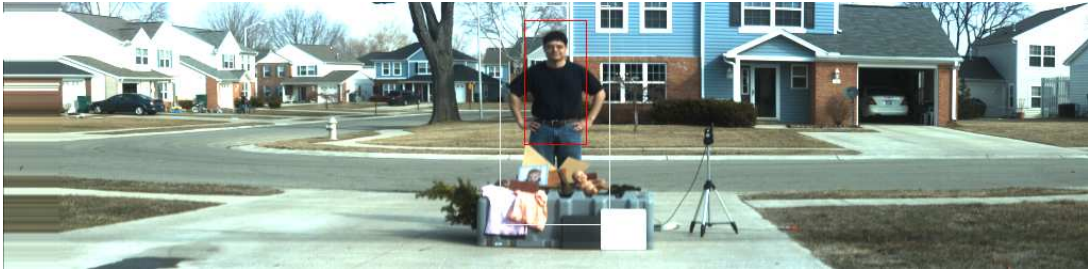(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.29:   Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 29. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

(c) Dismount detection boxes at 0.95 P_D operating point
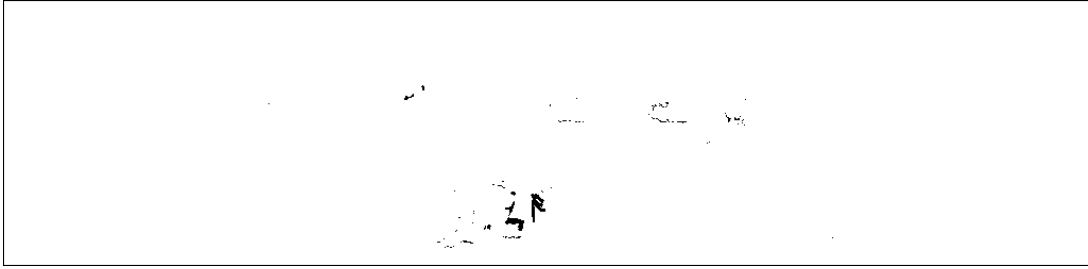

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.30:   Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 30. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

(c) Dismount detection boxes at 0.95 P_D operating point
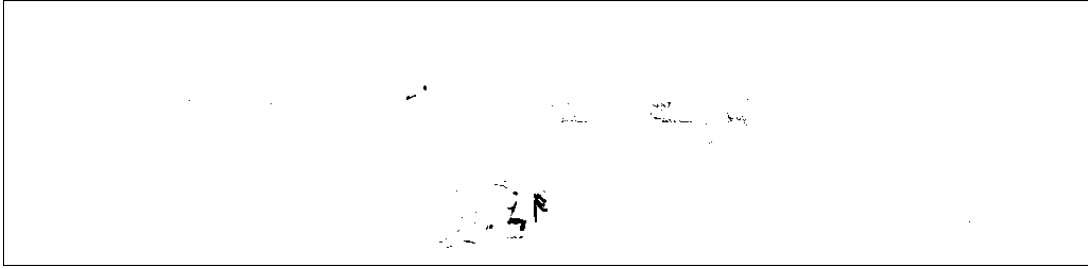

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.31:   Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 31. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

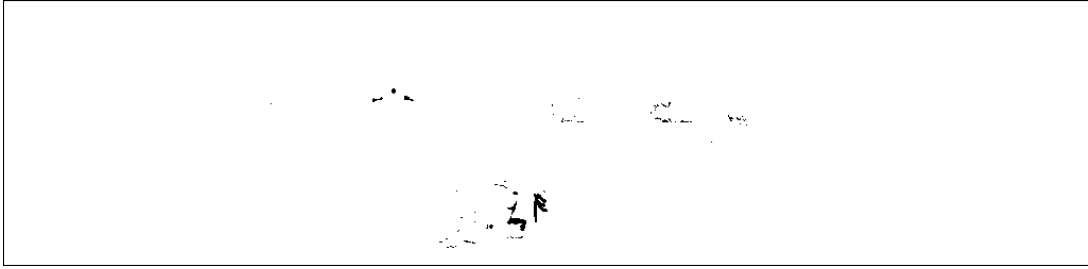(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.32:   Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 32. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.33:   Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 33. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.34:   Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 34. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

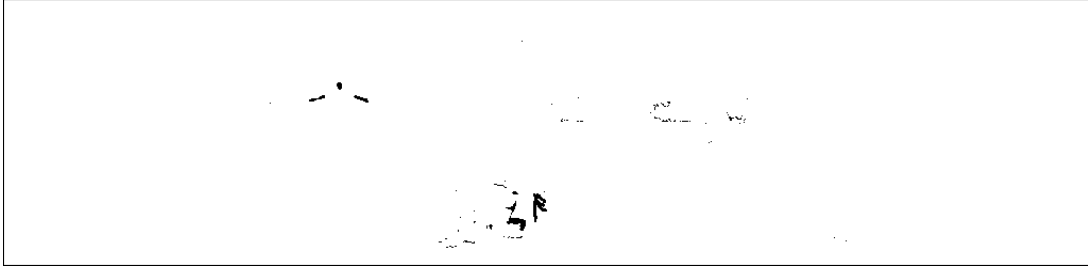(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.35:   Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 35. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.36: Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 36. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

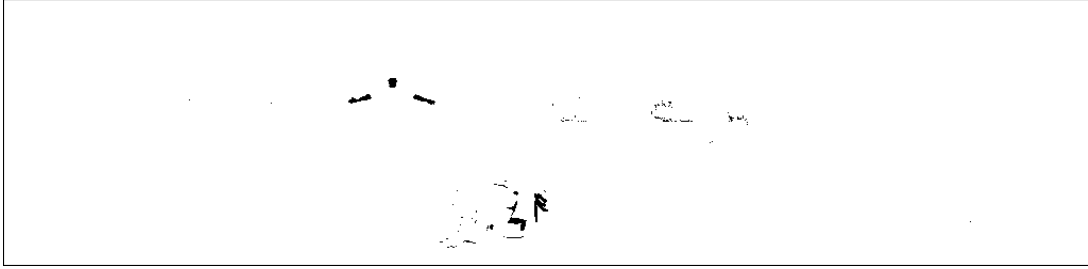(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.37:   Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 37. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image

(b) Skin detection mask

(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.38:   Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 38. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image
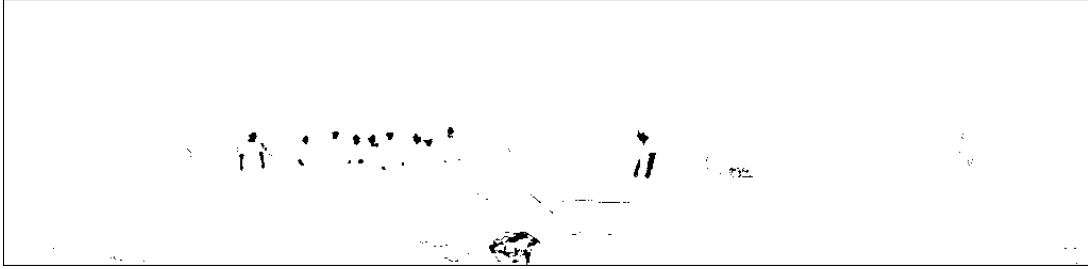
(b) Skin detection mask

(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.39:   Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 39. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image
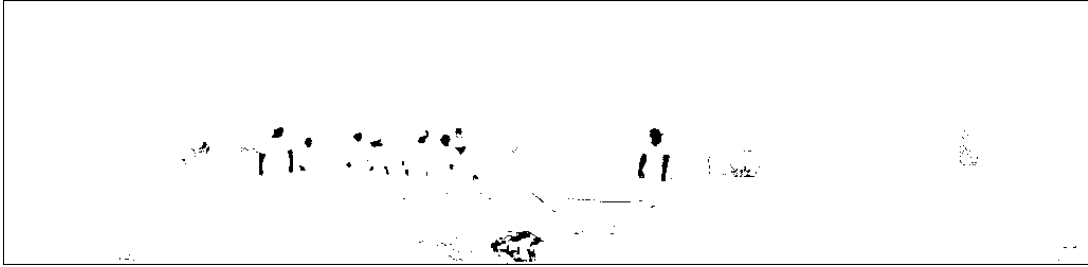
(b) Skin detection mask

(c) Dismount detection boxes at 0.95 P_D operating point

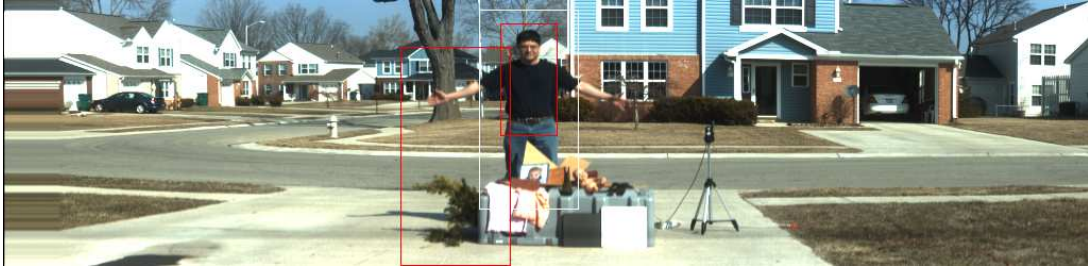(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.40:   Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 40. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image
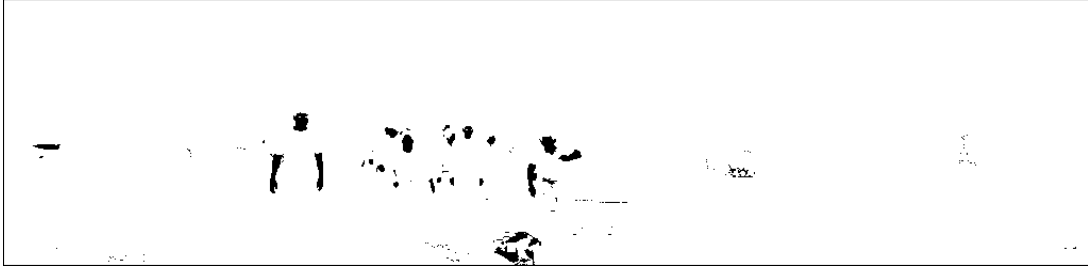
(b) Skin detection mask
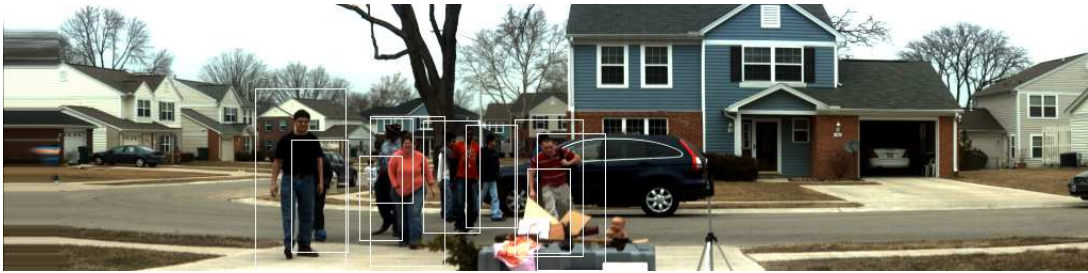
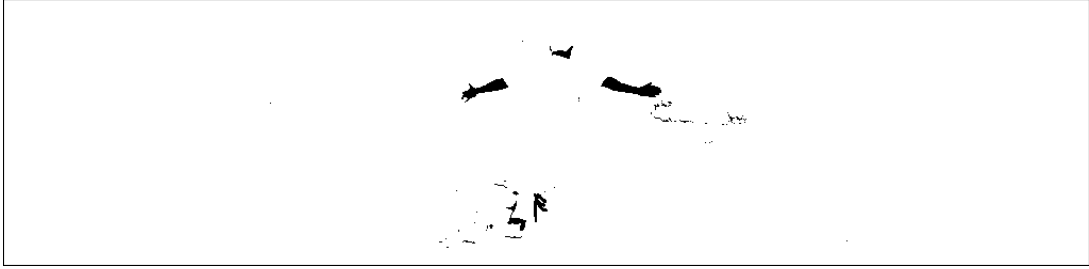(c) Dismount detection boxes at 0.95 P_D operating point

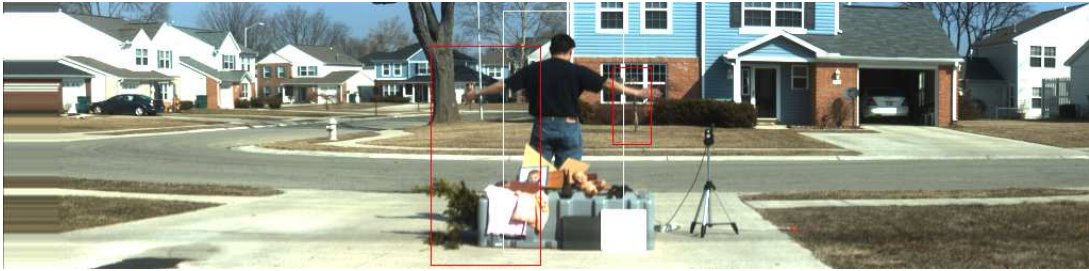(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.41:   Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 41. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

(a) Original HST3 image
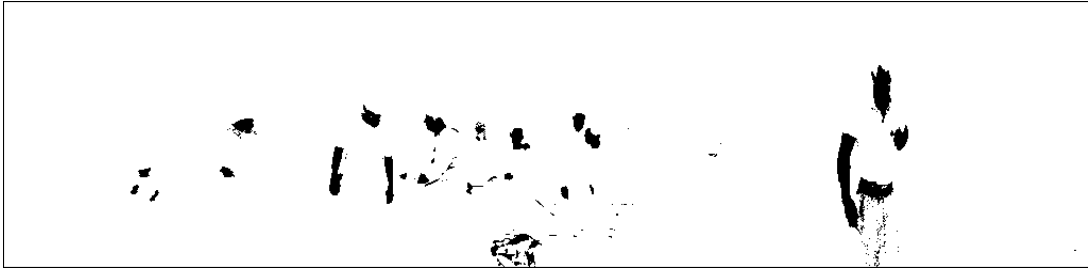
(b) Skin detection mask

(c) Dismount detection boxes at 0.95 P_D operating point

(d) Dismount detection boxes at 0.05 FPPF operating point

Figure B.42: Skin detection and skin-detection-cued HOG-based dismount detection results for HST3 image 42. (a) RGB conversion of the original HST3 image. (b) Rules-based skin detections ($\gamma \in \mathbb{R}[0.26, 0.93], \beta \in \mathbb{R}[-1, -0.02]$). (c) Skin-detection-cued HOG-based dismount detections at 0.95 P_D operating point. (d) Skin-detection-cued HOG-based dismount detections at 0.05 FPPF operating point. White boxes are hits. Red boxes are false alarms.

Appendix C. Likelihood Ratio Expectation Maximization Estimated Gaussian Mixture Model Parameters

This appendix includes example likelihood ratio parameter sets from each fold of the five-fold cross validation using the best-performing Monte-Carlo simulation. The parameters are presented for both the normalized difference green-red index (NDGRI) method and the normalized difference vegetation index (NDVI) method.

C.1 NDGRI Method

Table C.1:   NDGRI LRT Parameter Set 1.

Skin Distribution Parameters					
	Means		Standard Deviations		$\begin{bmatrix} a & b \\ b & c \end{bmatrix}$
Weights	NDGRI	NDSI	a	b	c
0.41682	-0.30437	0.69125	0.014384	-0.00044787	0.015709
0.10364	-0.47765	0.71174	0.041977	-0.0011302	0.016056
0.47954	-0.22926	0.55092	0.0074163	-0.00061235	0.0074237
Non-Skin Distribution Parameters					
	Means		Standard Deviations		$\begin{bmatrix} a & b \\ b & c \end{bmatrix}$
Weights	NDGRI	NDSI	a	b	c
0.2896	0.41809	0.34511	0.079366	-0.011888	0.019954
0.2104	-0.17539	0.087885	0.087202	-0.013372	0.017419
0.40539	-0.047265	0.26182	0.10632	-0.013049	0.18785
0.094609	0.06077	1	0.10716	-2.4547e-015	7.0916e-028

Table C.2:   NDGRI LRT Parameter Set 2.

Skin Distribution Parameters					
	Means		Standard Deviations $\begin{bmatrix} a & b \\ b & c \end{bmatrix}$		
Weights	NDGRI	NDSI	a	b	c
0.45667	-0.22813	0.54901	0.0071773	-0.0005665	0.007249
0.098022	-0.48695	0.71827	0.041252	-0.00017763	0.015986
0.44531	-0.30161	0.68493	0.014605	-0.00068813	0.015837
Non-Skin Distribution Parameters					
	Means		Standard Deviations $\begin{bmatrix} a & b \\ b & c \end{bmatrix}$		
Weights	NDGRI	NDSI	a	b	c
0.20867	-0.1738	0.089513	0.091738	-0.013903	0.016994
0.29133	0.41622	0.34997	0.080846	-0.011955	0.019433
0.37139	-0.040166	0.23453	0.1412	-0.0041843	0.20309
0.12861	0.0038624	0.88807	0.0066253	-0.00097311	0.016058

Table C.3:   NDGRI LRT Parameter Set 3.

Skin Distribution Parameters					
	Means		Standard Deviations $\begin{bmatrix} a & b \\ b & c \end{bmatrix}$		
Weights	NDGRI	NDSI	a	b	c
0.45342	-0.22837	0.54878	0.007262	-0.00058891	0.0071914
0.45028	-0.30138	0.68451	0.014503	-0.00065761	0.015901
0.096305	-0.49172	0.71826	0.040737	-0.00030859	0.016003
Non-Skin Distribution Parameters					
	Means		Standard Deviations $\begin{bmatrix} a & b \\ b & c \end{bmatrix}$		
Weights	NDGRI	NDSI	a	b	c
0.2964	0.41015	0.34468	0.078943	-0.010788	0.019589
0.2036	-0.18605	0.087626	0.088336	-0.014613	0.017634
0.13452	0.012277	0.89538	0.0067291	-0.00048387	0.014717
0.36548	-0.032222	0.23129	0.14569	-0.0032095	0.20029

Table C.4:   NDGRI LRT Parameter Set 4.

Skin Distribution Parameters					
	Means		Standard Deviations $\begin{bmatrix} a & b \\ b & c \end{bmatrix}$		
Weights	NDGRI	NDSI	a	b	c
0.4621	-0.22796	0.54941	0.0072343	-0.00055564	0.0072745
0.43932	-0.30283	0.68703	0.014525	-0.00057685	0.015799
0.098582	-0.48687	0.71382	0.041005	-0.00077207	0.016046
Non-Skin Distribution Parameters					
	Means		Standard Deviations $\begin{bmatrix} a & b \\ b & c \end{bmatrix}$		
Weights	NDGRI	NDSI	a	b	c
0.28875	0.41495	0.34782	0.078794	-0.012214	0.019571
0.21125	-0.16706	0.091292	0.091423	-0.012562	0.017085
0.37686	-0.037779	0.24156	0.14045	-0.004267	0.21107
0.12314	0.011227	0.88915	0.0057924	-9.0853e-005	0.015372

Table C.5:   NDGRI LRT Parameter Set 5.

	Means		Standard Deviations $\begin{bmatrix} a & b \\ b & c \end{bmatrix}$		
Skin Distribution Parameters					
Weights	NDGRI	NDSI	a	b	c
0.050133	-0.59208	0.7318	0.033064	0.0015989	0.016124
0.4286	-0.3185	0.70081	0.016378	0.00020274	0.015271
0.52127	-0.23059	0.55481	0.007588	-0.00060002	0.0078278
Non-Skin Distribution Parameters					
	Means		Standard Deviations $\begin{bmatrix} a & b \\ b & c \end{bmatrix}$		
Weights	NDGRI	NDSI	a	b	c
0.29662	0.40849	0.34461	0.080767	-0.011121	0.020191
0.20338	-0.18723	0.087516	0.086818	-0.014475	0.017403
0.3733	-0.04409	0.23317	0.1408	-0.0026442	0.1981
0.1267	0.0063058	0.89827	0.0058912	-0.00064135	0.013808

C.2   NDVI Method

Table C.6:   NDVI LRT Parameter Set 1.

	Means		Standard Deviations $\begin{bmatrix} a & b \\ b & c \end{bmatrix}$		
Skin Distribution Parameters					
Weights	NDVI	NDSI	a	b	c
0.15367	0.035487	0.66381	0.001974	0.00017854	0.012638
0.38571	0.24027	0.70751	0.013946	-0.0011796	0.015543
0.46063	0.18224	0.54568	0.0071414	-0.0013245	0.0073813
Non-Skin Distribution Parameters					
	Means		Standard Deviations $\begin{bmatrix} a & b \\ b & c \end{bmatrix}$		
Weights	NDVI	NDSI	a	b	c
0.30579	0.83431	0.32276	0.0095513	-0.0020667	0.023481
0.19421	0.39192	0.10894	0.038061	0.0058172	0.025447
0.11427	-0.11887	0.11237	0.38555	0.05796	0.4202
0.38573	0.067291	0.44727	0.038866	-0.032526	0.18267

Table C.7:   NDVI LRT Parameter Set 2.

Skin Distribution Parameters					
	Means		Standard Deviations $\begin{bmatrix} a & b \\ b & c \end{bmatrix}$		
Weights	NDVI	NDSI	a	b	c
0.38945	0.23958	0.70718	0.014	-0.0012945	0.01555
0.45621	0.18251	0.54512	0.0070739	-0.0012998	0.0073428
0.15433	0.035163	0.66205	0.001959	0.000207	0.012425
Non-Skin Distribution Parameters					
	Means		Standard Deviations $\begin{bmatrix} a & b \\ b & c \end{bmatrix}$		
Weights	NDVI	NDSI	a	b	c
0.30263	0.83598	0.32309	0.0090406	-0.001726	0.022803
0.19737	0.3928	0.11423	0.036388	0.0064582	0.027436
0.11368	-0.1514	0.12108	0.38669	0.068923	0.44034
0.38632	0.064141	0.47107	0.038873	-0.030889	0.17679

Table C.8:   NDVI LRT Parameter Set 3.

	Skin Distribution Parameters				
	Means		Standard Deviations $\begin{bmatrix} a & b \\ b & c \end{bmatrix}$		
Weights	NDVI	NDSI	a	b	c
0.48076	0.18635	0.54763	0.0070731	-0.0011553	0.0076632
0.35668	0.24298	0.71772	0.014339	-0.0018013	0.015123
0.16256	0.036812	0.65734	0.0020634	0.00011849	0.011971
	Non-Skin Distribution Parameters				
	Means		Standard Deviations $\begin{bmatrix} a & b \\ b & c \end{bmatrix}$		
Weights	NDVI	NDSI	a	b	c
0.29551	0.83708	0.32738	0.0089934	-0.0022423	0.022675
0.20449	0.40671	0.12022	0.041558	0.007967	0.027712
0.38633	0.071321	0.46782	0.039777	-0.031114	0.1754
0.11367	-0.15832	0.1766	0.36129	0.055615	0.42459

Table C.9:   NDVI LRT Parameter Set 4.

	Skin Distribution Parameters				
	Means		Standard Deviations $\begin{bmatrix} a & b \\ b & c \end{bmatrix}$		
Weights	NDVI	NDSI	a	b	c
0.4748	0.18603	0.5471	0.0070586	-0.0011658	0.0076018
0.36167	0.24268	0.71595	0.014265	-0.0016176	0.015166
0.16353	0.037025	0.65846	0.0020781	0.0001141	0.012115
	Non-Skin Distribution Parameters				
	Means		Standard Deviations $\begin{bmatrix} a & b \\ b & c \end{bmatrix}$		
Weights	NDVI	NDSI	a	b	c
0.20631	0.40917	0.11743	0.041941	0.0068124	0.025183
0.29369	0.8374	0.33145	0.0087117	-0.0026371	0.022916
0.39751	0.062235	0.45742	0.040581	-0.031246	0.18015
0.10249	-0.15489	0.18158	0.39856	0.062769	0.40457

Table C.10:   NDVI LRT Parameter Set 5.

Skin Distribution Parameters					
	Means		Standard Deviations $\begin{bmatrix} a & b \\ b & c \end{bmatrix}$		
Weights	NDVI	NDSI	a	b	c
0.15154	0.034884	0.66235	0.0019417	0.00018487	0.01248
0.38797	0.23959	0.70766	0.014091	-0.0013436	0.015527
0.46049	0.1826	0.54549	0.0071311	-0.0013438	0.0074363
Non-Skin Distribution Parameters					
	Means		Standard Deviations $\begin{bmatrix} a & b \\ b & c \end{bmatrix}$		
Weights	NDVI	NDSI	a	b	c
0.20184	0.39651	0.11678	0.038581	0.0069766	0.028383
0.29816	0.83654	0.32018	0.0089604	-0.0014221	0.022447
0.38842	0.062817	0.48182	0.039027	-0.03109	0.18391
0.11158	-0.16668	0.090277	0.37373	0.058886	0.45383

Bibliography

1. "ASD FieldSpec® 3 Website". URL http://www.asdi.com/products/fieldspec-3.

2. "Digital Imaging and Remote Sensing Image Generation (DIRSIG) Website". URL http://dirsig.cis.rit.edu/.

3. "MotionAnalysis® Eagle Digital RealTime System Website". URL http://www.motionanalysis.com/html/animation/eagle.html.

4. "Darpa Program: Combat Zones that See", 2003.

5. "Unites States Air Force Basic Doctrine, Air Force Doctrine Document 1", November 2003.

6. Agarwal, Shivani, Aatif Awan, and Dan Roth. "Learning to Detect Objects in Images via a Sparse, Part-Based Representation". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1475–1490, 2004.

7. Albiol, Alberto, Luis Torres, and Edward J. Delp. "Optimum color spaces for skin detection". *IEEE International Conference on Image Processing*, 1:122–124, October 2001.

8. Alonso, Ignacio P., David Fernández Llorca, Miguel Ángel Sotelo, Luis Miguel Bergasa, Pedro Revenga de Toro, Jesús Nuevo, Manuel Ocaña, and Miguel Aangel Garcia Garrido. "Combination of Feature Extraction Methods for SVM Pedestrian Detection". *IEEE Transactions on Intelligent Transportation Systems*, 8(2):292–307, 2007.

9. Anderson, R. and J. Parrish. "The optics of human skin". *Journal of Investigative Dermatology*, 77:13–19, 1981.

10. Bishop, Christopher M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1st ed. 2006. corr. 2nd printing edition, October 2007. ISBN 0387310738.

11. Blackburn, J., M. Mendenhall, A. Rice, P. Shelnutt, N. Soliman, and J. Vasquez. "Feature aided tracking with hyperspectral imagery". *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 6699 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*. August 2007.

12. Brand, Jason and John S. Mason. "A comparative assessment of three approaches to pixel-level human skin-detection". *Proceedings of 15th International Conference on Pattern Recognition*, 1:1056–1059, Sep 2000.

13. Breiman, Leo. "Bagging predictors". *Machine Learning*, 24(2):123–140, 1996.

14. Chen, Cheng-Yao and Wayne Wolf. "Background modeling and object tracking using multi-spectral sensors". *VSSN '06: Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*, 27–34. ACM, New York, NY, USA, 2006. ISBN 1-59593-496-0.

15. Clark, R., G. Swayze, R. Wise, E. Livo, T. Hoefen, R. Kokaly, and S. Sutley. "USGS digital spectral library splib06a: U.S. geological survey, digital data series 231". 2007. URL http://speclab.cr.usgs.gov/spectral.lib06.

16. Conaire, C., N. E. Oconnor, E. Cooke, and A. F. Smeaton. "Multispectral Object Segmentation and Retrieval in Surveillance Video". *International Conference on Image Processing*. 2006.

17. Cox, D.R. and D.V. Hinkley. *Theoretical Statistics*. 1974.

18. Dalal, N., B. Triggs, and C. Schmid. "Human Detection Using Oriented Histograms of Flow and Appearance". *Proceedings from European Conference on Computer Vision*, 428–441. 2006.

19. Dalal, Navneet and Bill Triggs. "Histograms of Oriented Gradients for Human Detection". Cordelia Schmid, Stefano Soatto, and Carlo Tomasi (editors), *International Conference on Computer Vision & Pattern Recognition*, volume 2, 886–893. INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, June 2005. URL http://lear.inrialpes.fr/pubs/2005/DT05.

20. Dowdall, Jonathan, Ioannis Pavlidis, and George Bebis. "Face Detection in the Near-IR Spectrum". *Proceedings of Infrared Technology and Applications XXIX*, 5074:745–756, 2003.

21. Duda, Richard O., Peter E. Hart, and David G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2 edition, November 2000.

22. Eismann, M. T. "OENG 633 Hyperspectral Remote Sensing", 2006. Air Force Institute of Technology course notes.

23. Eismann, M. T. "Strategies for hyperspectral target detection in complex background environments". *IEEE Aerospace Conference*, 10, 2006.

24. Enzweiler, M., P. Kanter, and D. M. Gavrila. "Monocular pedestrian recognition using motion parallax". *Proceedings of the IEEE Intelligent Vehicles Symposium*, 792–797. 2008.

25. Enzweiler, Markus and Dariu M. Gavrila. "Monocular pedestrian detection: survey and experiments". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2179–2195, December 2009.

26. Fitzpatrick, Thomas B., George Szabo, and Michael M. Wick. "Biochemistry and Physiology of Melanin Pigmentation". Lowell A. Goldsmith (editor), *Biochemistry and physiology of the skin*. Oxford University Press, New York, 1983.

27. Gavrila, D. M. and S. Munder. "Multi-cue pedestrian detection and tracking from a moving vehicle". *International Journal of Computer Vision*, 73(1):41–59, 2007.

28. Gomez, Giovani. "On selecting colour components for skin detection". *Proceedings of the International Conference on Pattern Recognition*, 16(2):961–964, 2002.

29. Haering, Niels C., Péter L. Venetianer, and Alan J. Lipton. "The evolution of video surveillance: an overview". *Machine Vision and Applications*, 19(5-6):279–290, 2008.

30. Han, Mei, Wei Xu, and Yihong Gong. "Video Foreground Segmentation Based on Sequential Feature Clustering". *Proceedings of the 18th International Conference on Pattern Recognition*, 492–496. IEEE Computer Society, Washington, DC, USA, 2006. ISBN 0-7695-2521-0.

31. Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer Series in Statistics. Springer, 2nd ed. 2009. corr. 3rd printing edition, September 2010. ISBN 0387848576.

32. Hsu, Rein-Lein, Mohamed Abdel-Mottaleb, and Anil K. Jain. "Face detection in color images". *IEEE International Conference on Image Processing*, 1:1046–1049, October 2001.

33. Jengo, C. and J. LaVeigne. "Sensor performance comparison of hyperspectir instruments 1 and 2". volume 3, 1805. March 2004.

34. Joachims, T. "Making large-Scale SVM Learning Practical". B. Schölkopf, C. Burges, and A. Smola (editors), *Advances in Kernel Methods - Support Vector Learning*, chapter 11, 169–184. MIT Press, Cambridge, MA, 1999.

35. Kilgore, George A. and P. Rand Whillock. "Skin Detection Sensor". *United States Patent Application Number 11/264,654*, October 2005.

36. Laboratory, Air Force Research. *Modtran4 version 3, revision 1*. Space Vehicles Directorate, Air Force Materiel Command, Hanscom AFB, Mass. 01731-3010, 2001. Tech. Rep.

37. Leibe, B., N. Cornelis, K. Cornelis, and L. Van Gool. "Dynamic 3D Scene Analysis from a Moving Vehicle". *IEEE Conference on Computer Vision and Pattern Recognition*, 1–8. 2007.

38. Leibe, Bastian, Edgar Seemann, and Bernt Schiele. "Pedestrian Detection in Crowded Scenes". *Proceedings from IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, 878–885. 2005.

39. Lowe, David G. "Distinctive Image Features from Scale-Invariant Keypoints". *International Journal of Computer Vision*, 60:91–110, 2004.

40. Manolakis, D. and G. Shaw. "Detection algorithms for hyspectral imaging applications". *IEEE Signal Processing Magazine*, 19(1):29–43, January 2002.

41. Marschner, Stephen R., Stephen H. Westin, Eric P. Lafortune, Kenneth E. Torrance, and Donald P. Greenberg. "Image-Based BRDF Measurement Including Human Skin". *Rendering Techniques*, 131–144. 1999.

42. McIvor, Alan M. "Background subtraction techniques", 2000. URL `citeseer.ist.psu.edu/mcivor00background.html`.

43. Mikolajczyk, Krystian, Cordelia Schmid, and Andrew Zisserman. "Human detection based on a probabilistic assembly of robust part detectors". *European Conference on Computer Vision*, volume I, 69–81. 2004. URL `http://lear.inrialpes.fr/pubs/2004/MSZ04`.

44. Miller, John L. *Principles of infrared technology: a practical guide to the state of the art.* first edition, 1994. ISBN 0442012101.

45. Miller, John L. and Edward Friedman. *Photonics rules of thumb: optics, electro-optics, fiber optics, and lasers.* McGraw-Hill Professional Publishing, 1996. ISBN 0070443297.

46. Mohan, A., C. Papageorgiou, and T. Poggio. "Example-Based Object Detection in Images by Components". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(4):349–361, April 2001.

47. Moon, T. K. "The expectation-maximization algorithm". *IEEE Signal Processing Magazine*, 13(6):47–60, 1996.

48. Munder, Stefan, Christoph Schnörr, and Dariu M. Gavrila. "Pedestrian Detection and Tracking Using a Mixture of View-Based Shape-Texture Models". *IEEE Transactions on Intelligent Transportation Systems*, 9(2):333–343, 2008.

49. Nakajima, Chikahito, Massimiliano Pontil, Bernd Heisele, and Tomaso Poggio. "Full-body person recognition system". *Pattern Recognition*, 36(9):1997–2006, 2003.

50. Nicodemus, Fred E. "Directional reflectance and emissivity of an opaque surface". *Applied Optics*, 4(7):767–773, 1965.

51. Nunez, Abel S. *A physical model of human skin and its application for search and rescue.* Ph.D. thesis, Air Force Institute of Technology, 2009.

52. Nunez, Abel S., Adam L. Brooks, Michael J. Mendenhall, and Richard K. Martin. "Human Skin Detection and Color Estimation in Hyperspectral Imagery". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. In draft.

53. Nunez, Abel S. and Michael J. Mendenhall. "Detection of human skin in near infrared hyperspectral imagery". *IEEE International Geoscience and Remote Sensing Symposium*. July 2008.

54. Nunez, Abel S., Michael J. Mendenhall, Heidi C. Bertram, and Adam L. Brooks. "Building an Integumentary System Hyperspectral Model for Avatars". *IEEE*

Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing, volume 1. August 2009.

55. Nunez, Abel S., Michael J. Mendenhall, Kevin C. Gross, and Heidi Bertram. "An engineering model of human skin in the visible and near infrared". *IEEE Transactions on Biomedical Engineering*, December 2009. In Draft.

56. Pal, Sankar K. and Amita Pal (editors). *Pattern Recognition From Classical to Modern Approaches*. World Scientific, 2002.

57. Papageorgiou, C. and T. Poggio. "A Trainable System for Object Detection". *International Journal of Computer Vision*, 38:15–33, 2000.

58. Pavlidis, I., P. Symosek, B. Fritz, M. Bazakos, and N. Papanikolopoulos. "Automatic detection of vehicle occupants: the imaging problem and its solution". *Machine Vision and Applications*, 11(6):313–320, May 2000.

59. Peer, Peter, Jure Kovac, and Franc Solina. "Human Skin Colour Clustering for Face Detection". URL `citeseer.ist.psu.edu/572745.html`.

60. Peskosky, Keith. *Design of a monocular multispectral skin detection, melanin estimation, and false alarm suppression system*. Master's thesis, Air Force Institute of Technology, March 2010.

61. Sabzmeydani, P. and G. Mori. "Detecting Pedestrians by Learning Shapelet Features". *IEEE International Conference on Compuer Vision and Pattern Recognition*. 2007.

62. Schölkopf, Bernhard and Alexander J. Smola. "Kernel-Machines.org software links". URL `http://www.kernel-machines.org/software`.

63. Schölkopf, Bernhard and Alexander J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. The MIT Press, 2002. ISBN 0262194759.

64. Seemann, Edgar, Mario Fritz, and Bernt Schiele. "Towards Robust Pedestrian Detection in Crowded Image Sequences." *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2007.

65. Shashua, Amnon, Yoram Gdalyahu, and Gaby Hayun. "Pedestrian Detection for Driving Assistance Systems: Single-frame Classification". *In Proceedings of IEEE Intelligent Vehicles Symposium*, 1–6. 2004.

66. Shet, V.D., J. Neumann, V. Ramesh, and L.S. Davis. "Bilattice-based Logical Reasoning for Human Detection". *IEEE Conference on Computer Vision and Pattern Recognition*, 1–8, June 2007.

67. Shimizu, Hiroaki. "Direction Estimation of Pedestrian from Multiple Still Images". *Proceedings from IEEE Intelligent Vehicles Symposium 2004*. 2004.

68. Sidenbladh, Hedvig and Michael J. Black. "Learning the Statistics of People in Images and Video". *International Journal of Computer Vision*, 54(1-3):181–207, 2003.

69. Soliman, N. A., M. J. Mendenhall, and J. R. Vasquez. "Spectral gating in hyperspectral-augmented target tracking". *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 6969 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*. May 2008.

70. Soliman, Neil A. *Hyperspectral-Augmented Target Tracking*. Master's thesis, Air Force Institute of Technology, March 2008.

71. Szarvas, M., A. Yoshizawa, M. Yamamoto, and J. Ogata. "Pedestrian Detection with Convolutional Neural Networks". *Proceedings from IEEE Intelligent Vehicles Symposium*, 223–228. 2005.

72. Theodoridis, Sergios and Konstantinos Koutroumbas. *Pattern Recognition, Third Edition*. Academic Press, February 2006. ISBN 0123695317.

73. United States Army Research Office. "Efficient and Robust Algorithms for Real-time Video Tracking of Multiple Moving Targets", 2007. Proposal.

74. Vezhnevets, V., V. Sazonov, and A. Andreeva. "A survey on pixel-based skin color detection techniques", 2003. URL `citeseer.ist.psu.edu/vezhnevets03survey.html`.

75. Viola, Paul, Michael J. Jones, and Daniel Snow. "Detecting Pedestrians Using Patterns of Motion and Appearance". *International Journal of Computer Vision*, 63(2):153–161, 2005. ISSN 0920-5691.

76. Wu, Bo and Ram Nevatia. "Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet based Part Detectors". *International Journal of Computer Vision*, 75(2):247–266, November 2007.

77. Yarbrough, Allan W., Michael J. Mendenhall, and Richard K. Martin. "The Effects of Atmospheric Mis-estimation on Hyperspectral-based Adaptive Matched Filter Target Detection as Measured by the Bhattacharyya Coefficient". *IEEE Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, volume 2. June 2010.

78. Zhang, Li, Bo Wu, and Ramakant Nevatia. "Detection and Tracking of Multiple Humans with Extensive Pose Articulation". *International Conference on Computer Vision*, 1–8. 2007.

79. Zhao, Tao and Ram Nevatia. "Tracking multiple humans in complex situations". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:1208–1221, 2004.

80. Zhu, Qiang, Shai Avidan, Mei C. Yeh, and Kwang T. Cheng. "Fast Human Detection Using a Cascade of Histograms of Oriented Gradients". *Proceedings*

from IEEE Conference on Computer Vision and Pattern Recognition, 1491–1498. IEEE Computer Society, 2006.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
26–03–2010	Master's Thesis	Sept 2008 — Mar 2010

4. TITLE AND SUBTITLE

Improved Multispectral Skin Detection and its
Application to Search Space Reduction for
Dismount Detection Based on
Histograms of Oriented Gradients

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Adam Lee Brooks, Capt, USAF

5d. PROJECT NUMBER

09ENG287

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GE/ENG/10-05

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Dr. Brian Tsou
711 Human Performance Wing
Information Operations and Special Programs Branch
2210 8th Street, Bldg. 164, WPAFB, OH 45433
(937) 255–8896; brian.tsou@wpafb.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/RHXB

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approval for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Due to the general shift from conventional warfare to terrorism and urban warfare by enemies of the United States in the late 20th Century, locating and tracking individuals of interest have become critically important. Dismount detection and tracking are vital to provide security and intelligence in both combat and homeland defense scenarios including base defense, combat search and rescue (CSAR), and border patrol. This thesis focuses on exploiting recent advances in skin detection research to reliably detect dismounts in a scene. To this end, a signal-plus-noise model is developed to map modeled skin spectra to the imaging response of an arbitrary sensor, enabling an in-depth exploration of multispectral features as they are encountered in the real world for improved skin detection. Knowledge of skin locations within an image is exploited to cue a robust dismount detection algorithm, significantly improving dismount detection performance and efficiency. This research explores multiple spectral features and detection algorithms to find the best features and algorithms for detecting skin in multispectral visible and short wave infrared (SWIR) imagery. This study concludes that using SWIR imagery for skin detection and color information for false alarm suppression results in 95% probability of skin detection at a false alarm rate of only 0.4%. Skin detections are utilized to cue a dismount detector based on histograms of oriented gradients. This technique reduces the search space by nearly 3 orders of magnitude compared to searching an entire image, while reducing the average number of false positives per image by nearly 2 orders of magnitude at 95% probability of dismount detection. The skin-detection-cued dismount detector developed in this thesis has the potential to make significant contribution to the United States Air Force human measurement and signature intelligence and CSAR missions.

15. SUBJECT TERMS

Skin Detection, Dismount Detection, Histograms of Oriented Gradients, Search and Rescue, Man-hunting.

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Maj Michael J. Mendenhall
U	U	U	UU	183	19b. TELEPHONE NUMBER *(include area code)* (937) 255–3636, ext 4614 Michael.Mendenhall@afit.edu

Standard Form 298 (Rev. 8–98)
Prescribed by ANSI Std. Z39.18