Automatic Target Cueing (ATC)

Prepared By:
Stephen Se & Mohsen Ghazel

MDA Systems Ltd.
13800 Commerce Parkway
Richmond, B.C., Canada, V6V 2J3

PWGSC Contract Number: W7701-135590/001/QCL

Contract Scientific Authority:
Mr. Philips Laou, Ph.D. 418-844-4000 x4218

The scientific or technical validity of this Contract Report is entirely the responsibility of the Contractor and the contents do not necessarily have the approval or endorsement of the Department of National Defence of Canada.

Contract Report
DRDC-RDDC-2014-C200
March 2014

Automatic Target Cueing (ATC)

Final Report

March 18, 2014

Stephen Se & Mohsen Ghazel

MDA Systems Ltd.

PWGSC Contract Title:	Automatic target cueing and facial recognition in visible & infrared spectrum for military operations
MDA Project Title:	6484 - ATC
MDA Document Number:	RX-RP-53-5696
Contract No.:	W7701-135590/001/QCL
Project Duration:	August 2 2013 – March 31 2014
DRDC Technical Authority:	Dr. Philips Laou (418) 844-4000 x4218

13800 Commerce Parkway
Richmond, B.C., Canada, V6V 2J3
Telephone (604) 278-3411
Fax (604) 231-2753

RESTRICTION ON USE, PUBLICATION, OR DISCLOSURE OF PROPRIETARY INFORMATION

The scientific or technical validity of this Contract Report is entirely the responsibility of the contractor and the contents do not necessarily have the approval or endorsement of Defence R&D Canada.

THIS PAGE INTENTIONALLY LEFT BLANK

Automatic Target Cueing (ATC)

Final Report

March 18, 2014

Stephen Se & Mohsen Ghazel

MDA Systems Ltd.
13800 Commerce Parkway
Richmond, BC, Canada
V6V 2J3

PWGSC Contract Title:	Automatic target cueing and facial recognition in visible & infrared spectrum for military operations
MDA Project Title:	6484 - ATC
MDA Document Number:	RX-RP-53-5696
Contract No.:	W7701-135590/001/QCL
Project Duration:	August 2 2013 – March 31 2014
DRDC Technical Authority:	Dr. Philips Laou (418) 844-4000 x4218

Prepared By: Stephen Se

Stephen Se March 18, 2014
Signature and Date

Prepared By: Mohsen Ghazel

MARCH 18, 2014
Signature and Date

Project Manager: Stephen Se

Stephen Se March 18, 2014
Signature and Date

MacDonald, Dettwiler and Associates Ltd
13800 Commerce Parkway
Richmond, BC, Canada
V6V 2J3

CHANGE RECORD

ISSUE	DATE	PAGE(S)	DESCRIPTION
1/0	Mar. 18, 2014	All	First Issue

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACRONYMS AND ABBREVIATIONS

ATC	Automatic Target Cueing
ATE	Assisted Target Engagement
BSD	Berkeley Software Distribution
CPU	Central Processing Unit
CVPR	Computer Vision and Pattern Recognition
DRDC	Defence Research and Development Canada
DVD	Digital Versatile Disk
FR	Facial Recognition
FSAR	Future Small Arms Research
GHz	Giga Hertz
HOG	Histograms of Oriented Gradients
Hz	Hertz
IEEE	Institute of Electrical and Electronics Engineers
IR	Infra-Red
LIBSVM	Library for Support Vector Machines
LSS	Local Self-Similarity
LW-IR	Long Wave Infra-Red
MDA	MDA Systems Ltd.
MEX	Matlab Executable
NA	Not Available
NORDIA	Non-Rigid Shape Analysis and Deformable Image Alignment
OpenCV	Open Source Computer Vision Library
PAMI	Pattern Analysis and Machine Intelligence
ROC	Receiver Operating Characteristic
SVM	Support Vector Machine
SW-IR	Short Wave Infra-Red

THIS PAGE INTENTIONALLY LEFT BLANK

1 INTRODUCTION

1.1 Background

Under the mandate of Future Small Arms Research (FSAR) program, Defence Research and Development Canada (DRDC) has been examining existing and future technologies for small arms capabilities with the objective of identifying technologies which could increase shot placement accuracy and reduce engagement time.

Automatic Target Cueing (ATC) is considered as one of the key enablers in future small arms technology. The goal is to assess feasibility and capability of Assisted Target Engagement (ATE) in small arms by combining ATC with electronic ignited ammunition and small arms weapons. It is believed that ATE may help improving shot placement and shortening engagement time in some situations.

1.2 Project Objectives

The objectives of this project are to conduct a feasibility study and provide software development support on Automatic Target Cueing (ATC) and Facial Recognition (FR) in visible and infra-red spectrum for military operations.

The project originally consisted of seven tasks, but two of the tasks were removed from the contract in December 2013. The project now consists of five tasks: Task 1 is related to literature survey of ATC and Task 2 is related to literature survey of FR, while the last three tasks are related to developing two chosen feature detectors and evaluating their performance on human target detection.

1.3 Scope

The literature survey for Tasks 1 and 2 has been completed and the findings are documented in [R-1] and [R-2] respectively.

This report fulfils the final report deliverable of this contract and contains the following elements:

- Description and implementation details of the feature detectors for person detection
- Performance evaluation of the feature detectors for person detection
- User's guide for the person detection tools

2 HUMAN TARGET DETECTION

Based on the Task 1 report [R-1], DRDC has selected two feature detectors for further investigation, namely Histogram of Oriented Gradients (HOG) [R-3] and Local Self-Similarity (LSS) [R-4]. The literature has shown promising results of person detection using these two detectors. This section provides a brief description and the implementation details of these two feature detectors.

2.1 Histogram of Oriented Gradients (HOG)

HOG [R-3] was first developed for pedestrian detection. As it operates on localized cells, it maintains invariance to geometric and photometric transformation except for orientation. Moreover, the use of coarse spatial sampling, fine orientation sampling and strong local photometric normalization allows individual body movement of pedestrians to be ignored as long as they maintain an upright position.

2.1.1 System Description

Figure 2-1 shows an overview of the implemented HOG SVM person detector. At run-time, person detection is handled by a binary Support Vector Machine (SVM) classifier, which is based on supervised learning. The SVM classifier has been pre-trained in OpenCV using typical pedestrian images from surveillance camera [R-5], in which the human subjects appear quite large. Also, the training data consists of visible-spectrum imagery but no Infra-Red (IR) imagery. While those default parameters can be used, the performance on FSAR data would not be optimal. Therefore, a training module is included to train the SVM classifier using representative FSAR data. Many positive and negative examples of persons are needed for training offline, and the resulting SVM parameters are then used for detection at run-time. The modules are described in more details below.

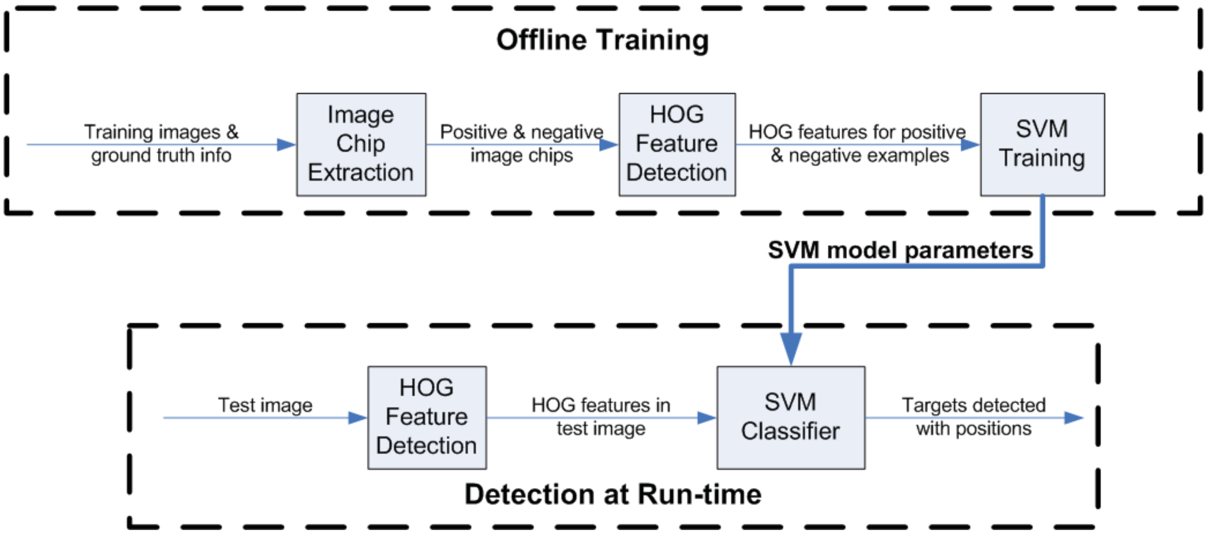

Offline Training
Training images & ground truth info
Image Chip Extraction
Positive & negative image chips
HOG Feature Detection
HOG features for positive & negative examples
SVM Training
SVM model parameters
Test image
HOG Feature Detection
HOG features in test image
SVM Classifier
Targets detected with positions
Detection at Run-time

Figure 2-1 HOG SVM Person Detector: Training and Detection

Image Chip Extraction: It is assumed that ground truth info is available for the selected training images. The ground truth info allows positive image chips to be extracted from the images automatically. Moreover, it also allows negative image chips to be extracted automatically, which are basically random image chips that do not overlap with the positive image chips. Optionally, the positive and negative image chips can be flipped sideways to double the number of training examples. For the DRDC-V LW-IR data set, image chips of 40x80 pixels are first extracted from the images based on the ground truth info provided by DRDC, and then re-sized as 48x96 windows. The window size depends on the size of the persons in the data set, e.g. a smaller size is used for the UK trial data set, as the targets are further away.

HOG Feature Detection: The image is uniformly divided into a dense grid of overlapping blocks with stride of 8 pixels. Each block was then split into 2x2 non-overlapping cells of size 8x8 pixels where HOGs were extracted using a histogram of 9 bins [R-3]. The object was encoded into a feature vector by concatenating the HOGs computed at each cells and blocks. A window size of 48x96 pixels would consist of 5 columns by 11 rows blocks, each block has 4 cells, and each cell has 9 bins, the resulting vector length is 5 x 11 x 4 x 9 = 1980. For example, a 640x480 image would contain 75 columns by 49 rows of 48x96 windows, i.e. 3675 vectors of 1980 length each. The HOG vector length is different for the UK trial data set, since a smaller window size is used.

SVM Training: Based on many known positive and negative examples and their HOG feature vectors, SVM training determines the optimal hyper-plane as a decision function. Some of the key SVM training parameters include the type of kernel function, gamma in the kernel function and the trade-off between training error and margin. The output is the SVM model parameters, which are fed into the SVM classifier. The effect of training examples will be examined in Section 3.

SVM Classifier: Using the SVM model parameters obtained from the training above as well as additional SVM detection parameters, the SVM classifier can make decisions regarding the presence of human in new test images, and also indicates their positions in the image.

In fact, this approach is not limited to human detection only. Depending on the training imagery, a different SVM classifier can be trained to detect different types of objects.

2.1.2 Implementation Details

We make use of the Open Source Computer Vision Library (OpenCV) [R-6], which is a comprehensive optimized C++ library for many computer vision algorithms. Both the HOG feature extraction and SVM classification are available in the OpenCV library.

We use the Mexopencv package [R-7], which is a collection of Matlab MEX functions that interface many OpenCV functions. It is suitable for prototyping OpenCV application in Matlab, using the optimized OpenCV as an external toolbox. Both the HOG extraction and SVM classification are available via the Mexopencv wrapper.

For SVM training, we use Library for Support Vector Machines (LIBSVM) package[R-8], which is a C++ library for SVM, including a Matlab interface for its C++ implementation.

The license information for these three third-party packages is shown in Table 2-1 below. They are all open-source packages and free of charge.

Table 2-1 Third-Party Packages License Information

Package	License
OpenCV [R-6]	BSD license – free for both academic and commercial use
Mexopencv [R-7]	BSD license – free for both academic and commercial use
LIBSVM [R-8]	BSD license – free for both academic and commercial use

Matlab scripts have been implemented for the followings:

- Extract positive image chips from images based on ground truth info
- Extract negative image chips from images based on ground truth info
- Output image chips to files
- Flip all the positive and negative examples to get more training examples
- Extract HOG features from the positive and negative image chips
- Train the SVM using the positive and negative HOG features to obtain new SVM model parameters
- Detect person by extracting HOG features and using the SVM classifier with the new SVM model parameters

Moreover, person detection with HOG and SVM has also been implemented in C++, using the OpenCV library directly, with slightly faster run-time performance. Both 32-bit and 64-bit pre-compiled executables are provided, where the 64-bit version runs slightly faster.

The following HOG SVM detection parameters are required for OpenCV:

- Hit threshold – threshold for the distance between features and the SVM classifying plane
- Scale – step size of scales to search
- Final threshold – score above this threshold is considered a detection

The sensitivity of the HOG SVM person detection results to the variation of these detection parameters will be examined in Section 3.

2.1.3 Preliminary Results

DRDC-V LW-IR data set contains 168 images with 308 positive examples, depicting images of human beings. For this preliminary test, 268 of them are selected for training. These images are flipped to increase the number of positive examples to 536. Two negative examples are randomly selected from each of the 168 images. These images are flipped to increase the number of negative examples to 672. SVM training on these positive and negative examples generates a set of new SVM model parameters, which are used for subsequent detection. Figure 2-2 shows the person detection results on several images that were not used for training. The results are promising as persons are correctly detected with no false alarms.

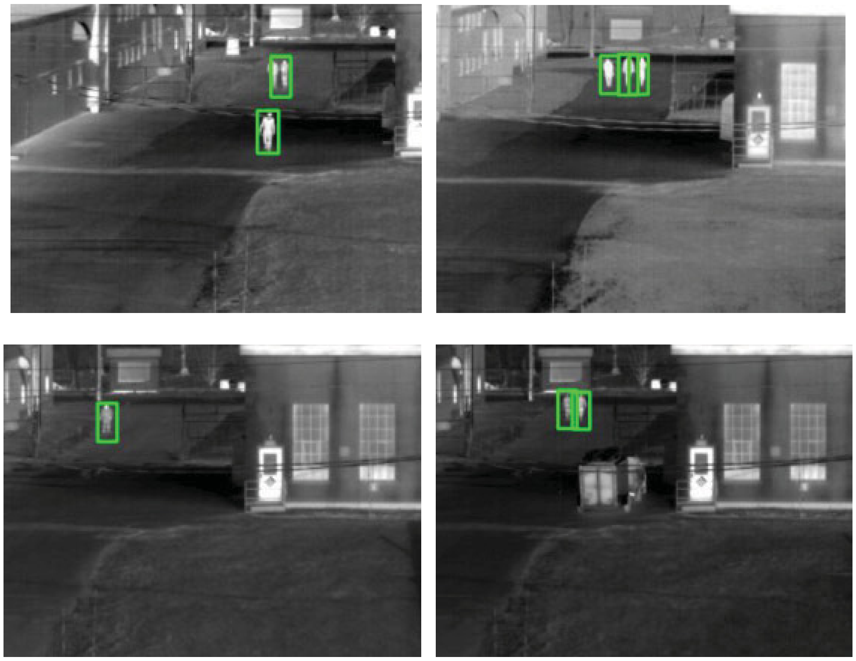


Figure 2-2 Examples of Human Detection with HOG SVM Detector – Parameters (0.4, 1.05, 0.0)

2.2 Local Self-Similarity (LSS)

[R-4] presents an approach for matching similar shapes between images/videos, based on the LSS features. By matching an ensemble of LSS descriptor vectors, it can detect objects in cluttered images well. This is a general shape matching approach that looks for similar shapes based on a query image example, and is not limited to human shapes.

2.2.1 System Description

Figure 2-3 shows an overview of the implemented LSS shape matching system. Given a single example query image of an object of interest, the goal is to find the position, both spatially and in scale, of all instances of that object in the test image. The modules are described in more details below.

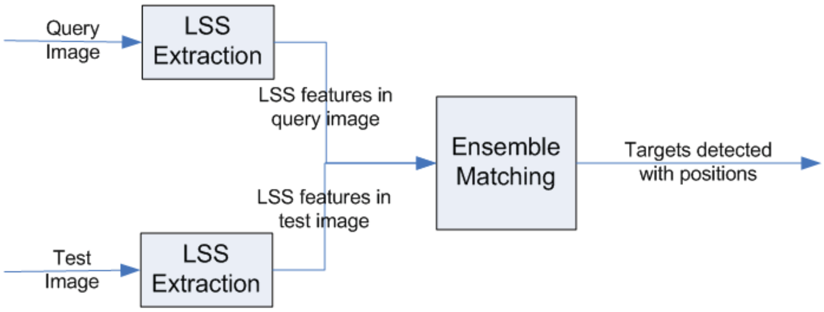

Query Image
LSS Extraction
LSS features in query image
Ensemble Matching
Targets detected with positions
LSS features in test image
Test Image
LSS Extraction

Figure 2-3 LSS Shape Matching System

LSS Extraction: The LSS descriptor is measured densely throughout the image at multiple scales, while accounting for local and global geometric distortions. A 5×5 image patch centered at pixel *q(x, y)* is correlated with a larger surrounding image region of diameter 41 pixels, resulting in a local internal correlation surface. The correlation surface is then transformed into a log-polar representation, with 80 bins (20 angles, 4 radial intervals). The maximal values in those bins form the 80 entries of the descriptor vector, which is then normalized. The number of LSS vectors in the image depends on how dense the descriptor is to be extracted. If it is extracted at every pixel, both the extraction and matching could take a long time due to processing the large number of descriptors.

Ensemble Matching: The basic approach to ensemble matching is a generalized Hough Transform, where each matched feature votes on the object centre, as described in [R-9]. For each LSS feature in the query image, the best matches are found in the test image based on Euclidean distance between the descriptors. Each match is allowed to vote for a possible position of the object centre in a 2D Hough space. Matching over scales is facilitated by performing this centre-voting procedure multiple times, in each case scaling the transform by an appropriate scale factor. The votes are clustered across scales, and the highest peaks with the largest number of supporting matches are chosen as the best matches.

2.2.2 Implementation Details

Same as above, we make use of the OpenCV [R-6] and Mexopencv package [R-7]. LSS extraction is available in OpenCV but not available in Mexopencv, therefore, a MEX wrapper has been implemented to add this function to the Mexopencv interface. This allows invoking OpenCV's SelfSimDescriptor function in Matlab. LSS descriptors are not computed around the image border (22 pixels), due to the correlation surface window size. The LSS ensemble matching is not available in OpenCV, and hence we implemented it.

Matlab script has been implemented for the followings:

- Extract and match LSS between query image and test image, to find instances of query image in the test image across multiple scales. It makes use of OpenCV's descriptor matcher class which can efficiently find the nearest neighbours for each descriptor.
- A batch mode where it can handle a set of query images, to find instances of any of the query images in the test image across multiple scales.

Moreover, LSS extraction and matching for one query image has also been implemented in C++, using the OpenCV library directly, with faster run-time performance. Both 32-bit and 64-bit pre-compiled executables are provided, where the 64-bit version runs slightly faster.

The key LSS matching parameters include:

- Win Stride – stride (in pixels) for how dense the uniform sampling is to extract the LSS descriptors
- Hough bin size – bin size (in pixels) for the Hough Transform
- Score threshold – score above this threshold is considered a match. The score is computed after clustering, as the product between the ratio of votes to average votes and the ratio of supporting matches to average matches.

The sensitivity of the LSS person detection results to the variation of these matching parameters will be examined in Section 3.

2.2.3 Preliminary Results

Query images are needed for LSS matching. Query images are cropped from one of the DRDC-V LW-IR image, and used as templates to search in other images. Figure 2-4 shows two examples of query images and the LSS matching results. The results are promising as correct instances of the query image are found with no false alarms.

The effects of different query images and the sensitivity of the LSS matching will be examined in Section 3.

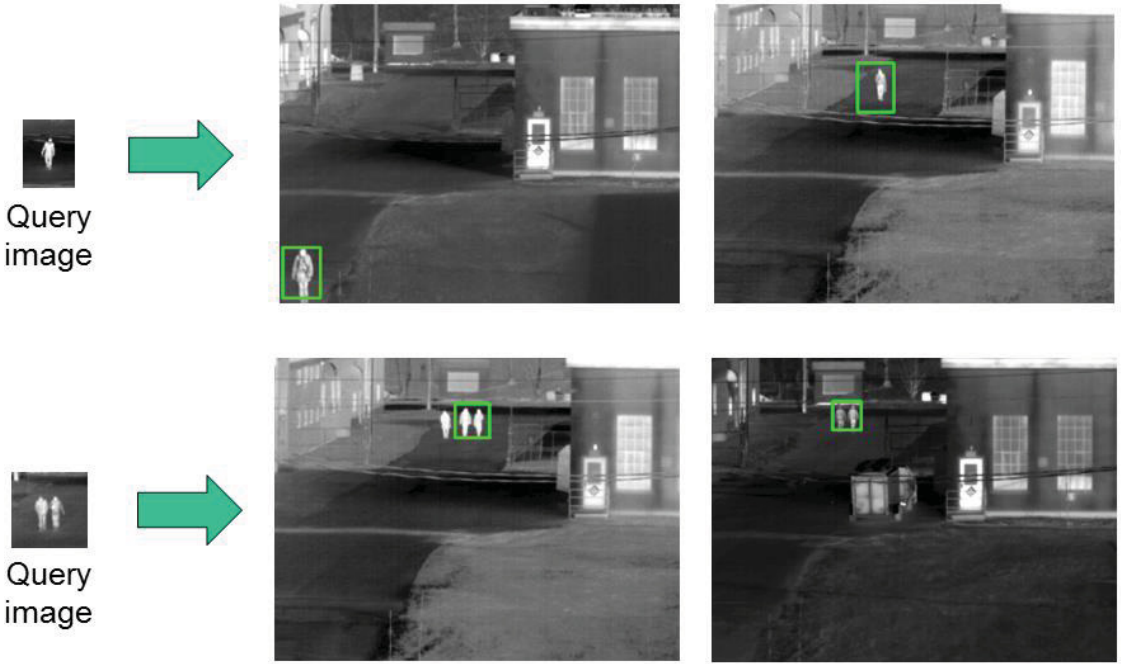


Figure 2-4 Examples of Human Detection with LSS Matching – Parameters (4, 4, 3)

THIS PAGE INTENTIONALLY LEFT BLANK

3 PERFORMANCE EVALUATION

In this section, we examine the sensitivity of the HOG SVM and the LSS person detection algorithms' results to the variation of some of their respective detection parameters. Typically, there is a trade-off between the probability of true person detection and the rate of false alarm, while adjusting some of the key algorithms' parameters. For example, for the HOG SVM algorithm, reducing the hit threshold should increase the probability of detection but may also increase the false alarm rate. Therefore, a more detailed experimentation and sensitivity analysis of the key parameters of the HOG SVM and LSS algorithms are needed in order to select more "optimized" values for their input parameters and attain sufficiently high detection probability, while keeping the false alarm rate low.

The sensitivity analysis presented in this section is based on the Receiver Operating Characteristic (ROC) curves. A ROC curve is a graphical plot which illustrates the performance of a binary classifier system as one (or more) of its discrimination parameters is (are) varied. It is created by plotting the probability of true detection as a function of the false alarm rate, at various threshold settings. In this work, the false alarm rate is defined as the average number of false detections per processed image.

The input data for the undertaken performance evaluation is summarized in Table 3-1. Due to differences between their acquisition properties, such as differences in imaging systems and sensors, range, field of view and quality of the collected imagery, the various data sets are processed separately.

Comprehensive results will be presented for the DRDC-V LW-IR data set, while only the key results will be included for the other data sets. We begin with illustrating the detection results for the DRDC-V LW-IR data set generated using the HOG SVM and LSS algorithms.

Table 3-1 Input Data Sets

#	Data set	Sensor & Resolut-ion	Acquisition date	Folder	Number of images	Imagery IDs	Ground truth file name	Ground truth structure name
1	DRDC-V trial	LW-IR (640x480)	NA	DRDC_ IR_Data	168	001 - 168	groundtruth.mat	info
2	UK trial	LW-IR (640x480)	04Dec2013	d1	194	1 - 194	info_dec_d.mat	4_dec_d1
				d2	202	1 - 202		4_dec_d2
			05Dec2013	d1	266	1 - 266		5_dec_d1
				d2	115	1 - 115		5_dec_d2
3		SW-IR (320x256)	04Dec2013	s1	144	1 - 144	info_dec_S.mat	4_dec_s1
				s2	102	1 - 102		4_dec_s2
				s3	95	1 - 95		4_dec_s3
				s4	122	1 - 122		4_dec_s4
				s5	140	1 - 140		4_dec_s5
				s6	157	1 - 157		4_dec_s6
			05Dec2013	s1	162	1 - 162		5_dec_s1
				s2	68	1 - 68		5_dec_s2
4	DRDC-V trial	Visible (1023x767)	NA	DRDC_ Vis_Data	168	001 – 201 (some missing images)	info_vis1_fix.mat	info_vis1_fix

3.1 DRDC-V LW-IR Data

3.1.1 HOG SVM

3.1.1.1 Selected Parameters

We examine the sensitivity of the HOG SVM person detection results to the variability of the following HOG input parameters:

- Hit threshold – threshold for the distance between features and the SVM classifying plane
- Scale – step size of scales to search
- Final threshold – score above this threshold is considered a detection

Since HOG SVM requires training, we first discuss how each data set has been split into training and test subsets.

3.1.1.2 SVM Training Data

For each data set, a subset of the images is selected for training the SVM classifier. The SVM training is based on manually selected ground truth from the training subset of images. For each image in the training set, we select the followings:

- Positive examples: All persons in the image
- Negative examples: Two examples are randomly selected that do not overlap with each other or with the positive examples.

In an effort to increase the number of positive and negative exemplars, we also flip each exemplar image, with respect to vertical central axis, resulting in twice as many positive and negative exemplars. Table 3-2 illustrates the training imagery for DRDC-V LW-IR data set.

Table 3-2 SVM Training Data for DRDC-V LW-IR Data

#	Training	Number of images	Imagery IDs	# of Training Examples
1	Positive examples	141	027 - 168	2×268 = 536
2	Negative examples	168	001 - 168	2×2×168 = 672

Generally, the training and test images subsets should be mutually exclusive. However, because the DRDC-V LW-IR data set is a relatively small data set, all images are processed, including those used for training.

3.1.1.3 ROC Curves

We generate ROC curves as well as other plots illustrating the sensitivity of the detection results as well as the computational complexity to the variations of the key HOG SVM parameters discussed above.

Table 3-3 illustrates the range of variation for each key HOG SVM detection parameter and the combination of the parameters' values yielding the best experimental detection results. The best detection results are defined in terms of achieving the highest probability of detection, without taking the false alarm rate into consideration.

Figure 3-1 illustrates the ROC curve resulting from varying the HOG SVM key parameters, as specified in Table 3-3. The best result in terms of highest probability of detection yields a probability of detection of about 62% at a false alarm rate of about 2.8 false detections per frame.

Table 3-3 Variability Ranges of HOG SVM Detection Parameters for DRDC-V LW-IR Data

HOG SVM Parameter	Minimum Value	Step Size	Maximum Value	Best Result Combination
Hit-threshold	0.0	0.5	5.0	0.0
Scale	0.0	0.05	2.0	1.05
Final-threshold	0.0	1.0	10.0	0.0
			Probability of detection	0.6201
			False alarm rate	2.7917

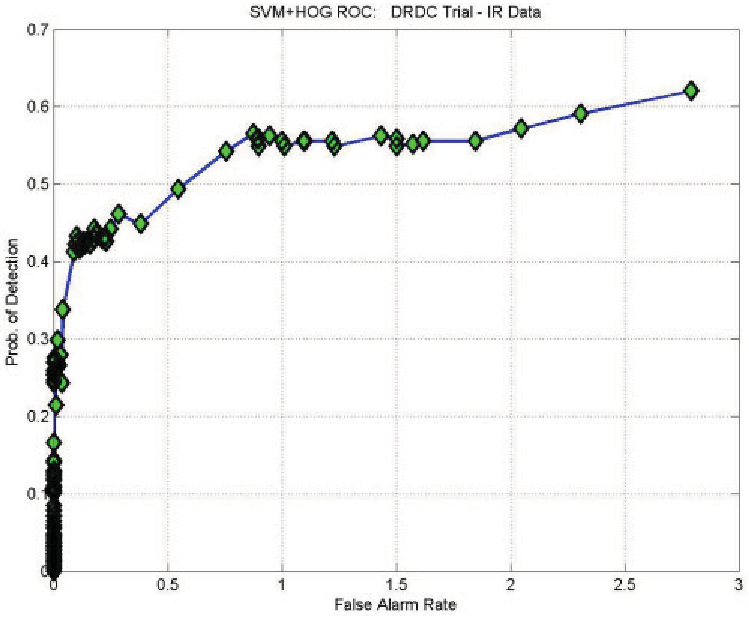


Figure 3-1 ROC Generated by Varying HOG SVM Detection Parameters for DRDC-V LW-IR Data

The combination of input parameters yielding the best detection results is used to illustrate the sensitivity of the detection and false alarm rates with respect to the variation of the input parameters according to the following two types of joint variations:

- Vary one parameter while fixing the other two parameters to their best-results values
- Vary two parameters while fixing the third parameter to its best-results value.

Figure 3-2, Figure 3-3 and Figure 3-4 illustrate the sensitivity of the false alarm rate and probability of detection to the variation of the hit threshold, scale and final-threshold, respectively. In each case, the indicated HOG SVM parameter is varied while the other two parameters are fixed to their respective values yielding the best detection results, as specified in Table 3-3. As expected, the detection probability and false alarm rate decay rapidly as the hit and final threshold parameters' increase, while the scale parameter has a distinct experimental optimal value of 1.05.

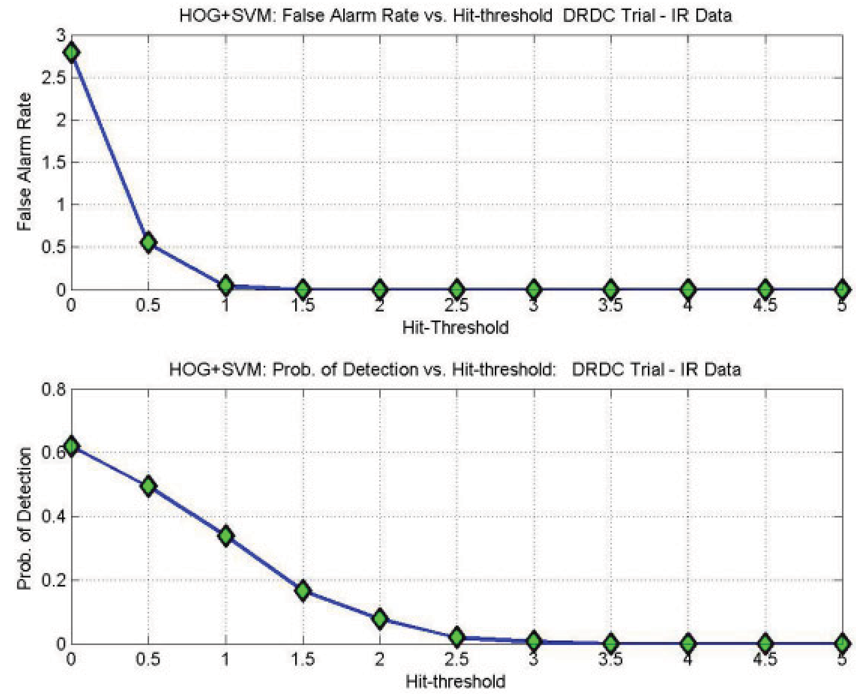


Figure 3-2 Sensitivity of Results to Variation of Hit-Threshold Only for DRDC-V LW-IR Data

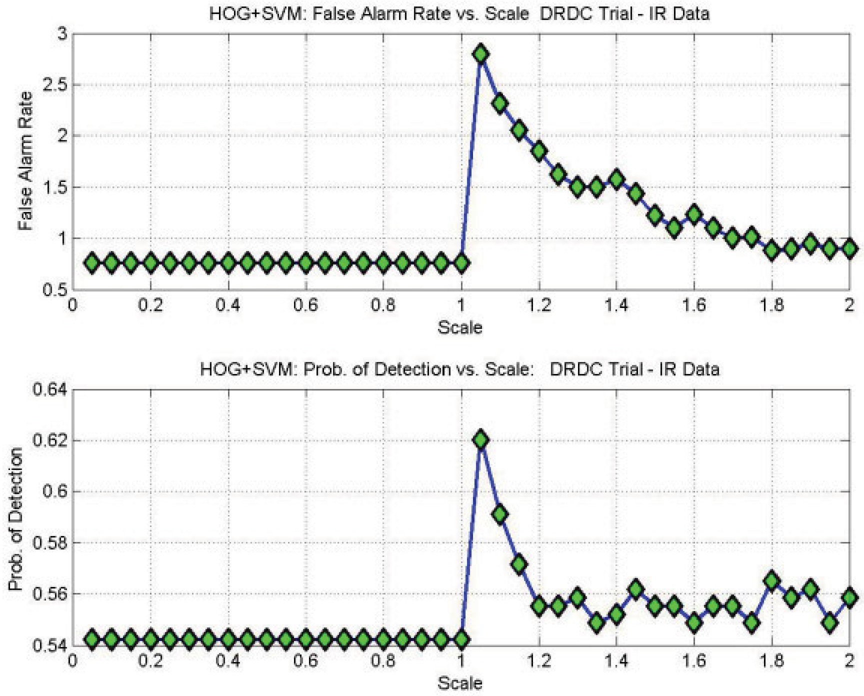


Figure 3-3 Sensitivity of Results to Variation of Scale Only for DRDC-V LW-IR Data

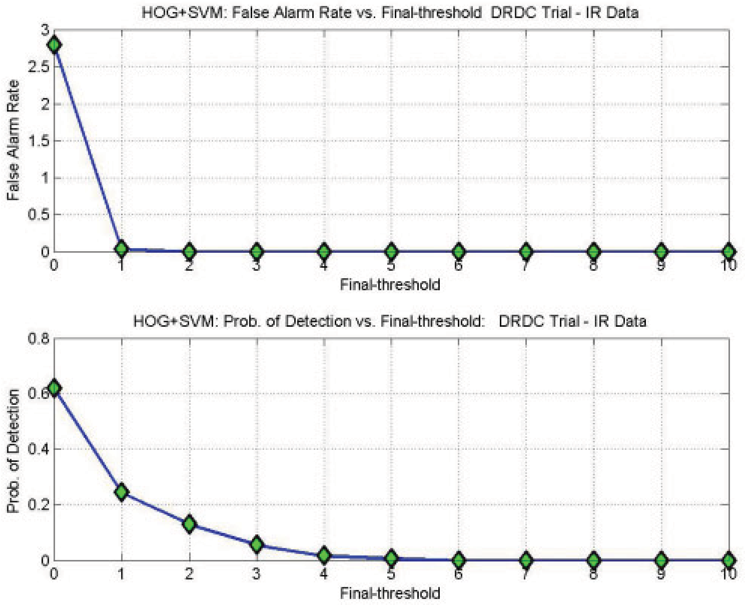


Figure 3-4 Sensitivity of Results to Variation of Final-Threshold Only for DRDC-V LW-IR Data

Figure 3-5, Figure 3-6 and Figure 3-7 illustrate the sensitivity of the false alarm rate and probability of detection to the simultaneous variation of two HOG SVM detection parameters, while keeping the third parameter fixed to its value yielding the best detection results, as specified in Table 3-3. It is clear that the detection results are most sensitive to the scale parameter, yielding a global optimum at 1.05. As such, we shall fix the scale parameter to this optimal value for the remaining HOG SVM ROC performance analysis in this report.

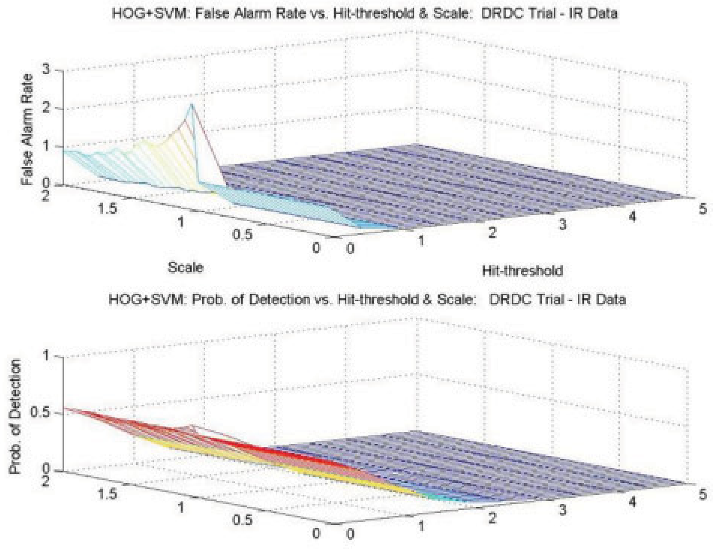


Figure 3-5 Sensitivity of Results to Variation of Hit-Threshold and Scale Only for DRDC-V LW-IR Data

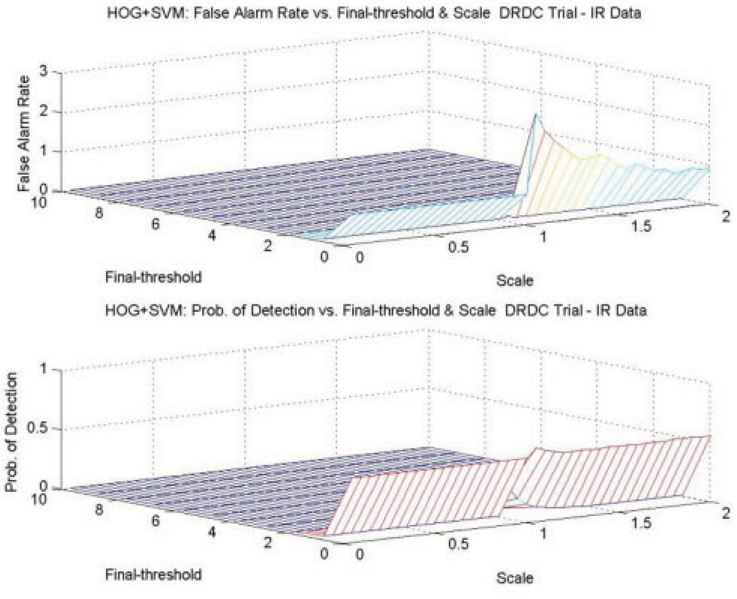


Figure 3-6 Sensitivity of Results to Variation of Final-Threshold and Scale Only for DRDC-V LW-IR Data

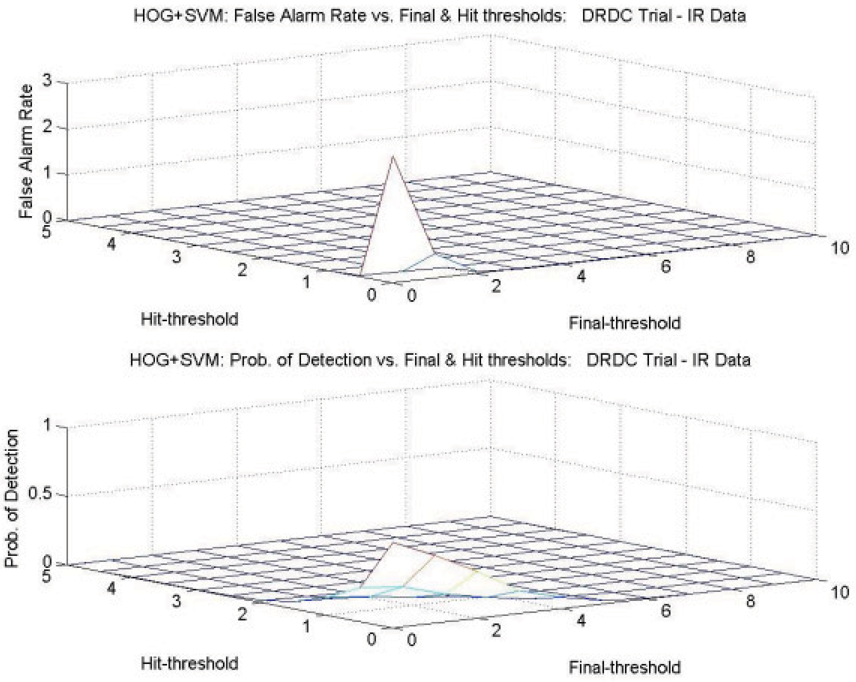


Figure 3-7 Sensitivity of Results to Variation of Hit-Threshold and Final-Threshold Only for DRDC-V LW-IR Data

3.1.1.4 Processing Time

Figure 3-8, Figure 3-9 and Figure 3-10 illustrate the dependence of the processing time on the HOG SVM detection parameters. We note that the optimal scale parameter value of 1.05 yields the most computationally expensive results of about 0.2 sec/frame (i.e. 5 Hz) on Intel Xeon W3520 2.67GHz CPU. The computational complexity is less sensitive to the variations of the hit and final threshold parameters, especially for sufficiently large values of these parameters. This is expected since the number of detections decreases rapidly as the hit and final-threshold parameters increase.

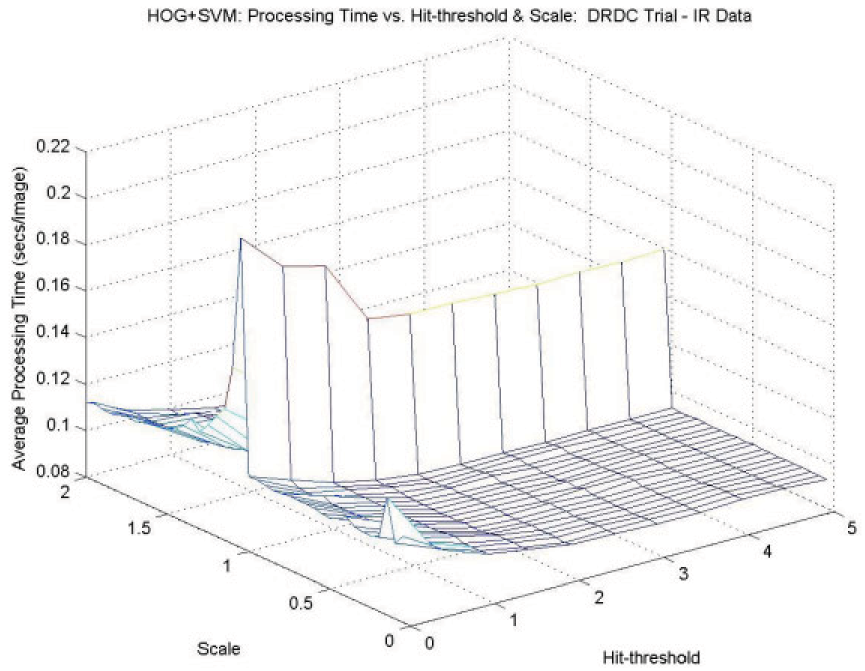


Figure 3-8 Processing Time Dependence on Variation of Hit-Threshold and Scale for DRDC-V LW-IR Data

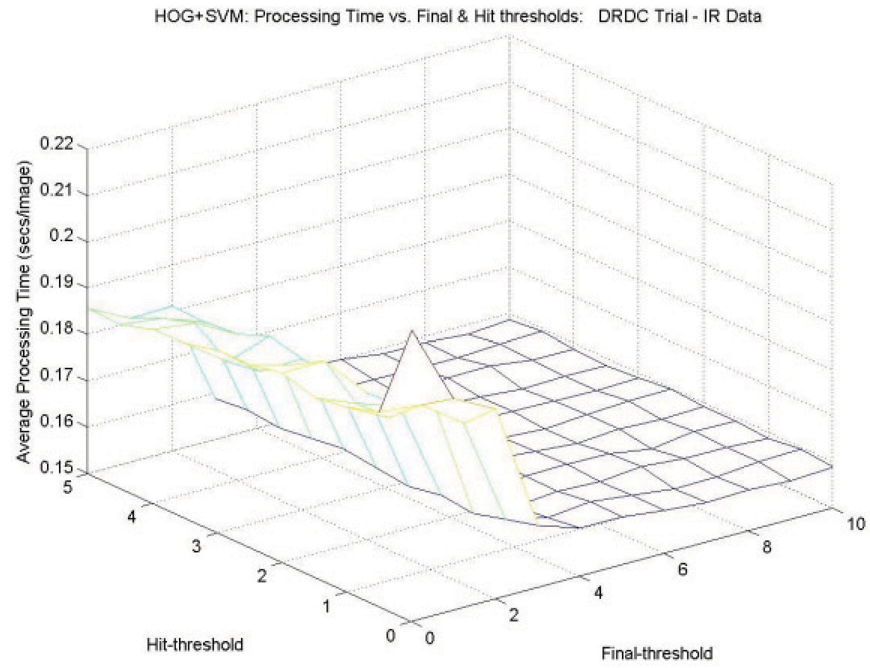


Figure 3-9 Processing Time Dependence on Variation of Hit-Threshold and Final-Threshold for DRDC-V LW-IR Data

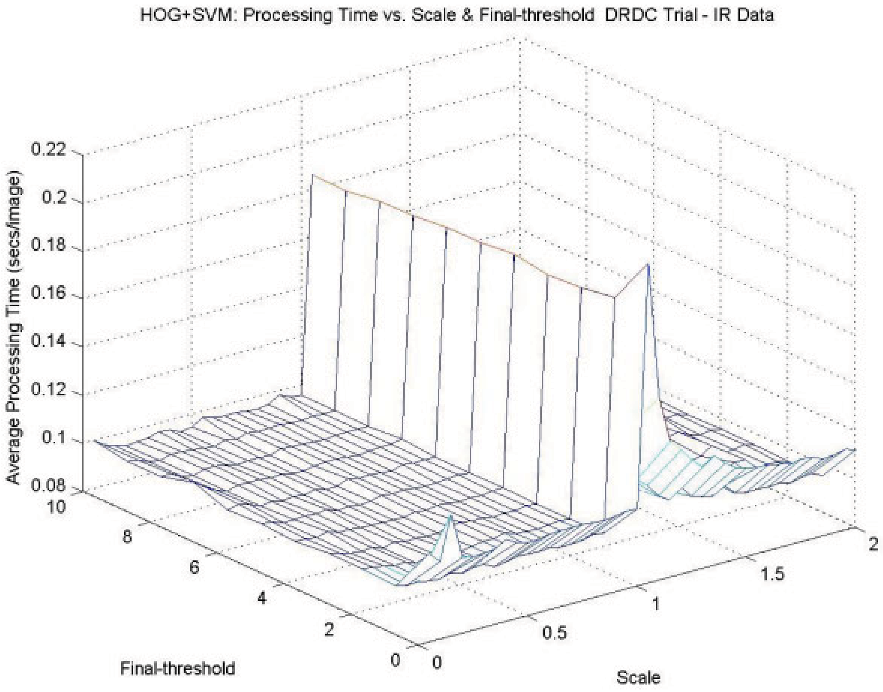


Figure 3-10 Processing Time Dependence on Variation of Final-Threshold and Scale for DRDC-V LW-IR Data

3.1.2 LSS Matching

Next, we illustrate and discuss the LSS matching results for the DRDC-V LW-IR data set.

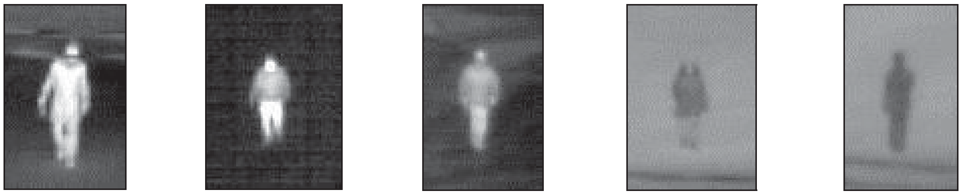
3.1.2.1 Selected Parameters

We examine the sensitivity of the LSS person detection results to the variability of the following key LSS matching parameters:

- Win Stride – stride (in pixels) for how dense the uniform sampling is to extract the LSS descriptors
- Hough bin size – bin size (in pixels) for the Hough Transform
- Score threshold – score above this threshold is considered a match. The score is computed after clustering, as the product between the ratio of votes to average votes and the ratio of supporting matches to average matches.

3.1.2.2 Query Images

As discussed in Section 2.2, the LSS algorithm is a shape-matching approach, which attempts to match the query images to similar shapes in the test image. For the purpose of person detection, query images of human beings are required and Figure 3-11 illustrates the query images used for DRDC-V LW-IR data set. The index of the image from which the each query sub-image has been extracted is also indicated. An effort was made to select different types of person's templates with distinct shapes, sizes and poses.



(a) 001 (b) 020 (c) 045 (d) 070 (e) 150

Figure 3-11 Selected Query Images for DRDC-V LW-IR Data

When multiple query images are used, there are different approaches related to how to handle the images and combine their results:

1. One approach is to invoke the LSS algorithm using one query image at a time and then accumulate the detection results (true detections and false alarms) stemming from the different query images, which are processed independently. The main disadvantage of this approach is that the number of false alarms can be over-estimated, as different query images may point to the same false alarms. Adding these duplicate false alarms resulting from different query images leads to over-estimating the number of false alarms. Another disadvantage of this approach is that the test image will have to be processed multiple times for each query image, and hence the LSS descriptors of the test image are re-computed repeatedly. This results in significant redundancy and increase in computational complexity and processing time.
2. A better approach is to invoke the LSS algorithm with all the query images. The LSS algorithm will still match each query image independently, however it would identify duplicate detections and ensure they are only counted once. Duplicate detections are identified based on a minimum percentage of overlap (80%) between their respective detected bounding boxes. This percentage overlap threshold is a parameter that can be adjusted as desired. Another key advantage of this approach is that the LSS descriptors for the test image is computed only once and then matched to the descriptors of the different query images. This results in significant reduction in the computational complexity and processing time.

In this work, we adopted the second approach of invoking LSS with all the query images (referred to as the batch mode in the User's Guide) because it is less computational expensive and yields more accurate estimate of the false alarm rate without compromising the probability of detection.

3.1.2.3 ROC Curves

We generate ROC curves as well as other plots depicting the sensitivity of the detection results and computational complexity to the variations of the key LSS parameters discussed above.

Table 3-4 illustrates the range of variation for each key LSS detection parameter and the combination of the parameters' values yielding the best detection results. As for the HOG SVM results, the best detection results are defined in terms of achieving the highest probability of detection, without taking the false alarm rate into consideration.

Figure 3-12 illustrates the ROC curve resulting from the varying the LSS key parameters, as specified in Table 3-4. The "best" result in terms of highest probability of detection yields a probability of detection of about 61.5% at a false alarm rate of about 6.5 false detections per frame.

Table 3-4 Variability Ranges of LSS Detection Parameters for DRDC-V LW-IR Data

LSS Parameter	Minimum Value	Step Size	Maximum Value	Best Result Combination
Window-stride	2	2	8	2
Hough bin-size	4	4	8	4
Score threshold	1.5	0.5	4.0	1.5
			Probability of detection	0.6169
			False alarm rate	6.4821

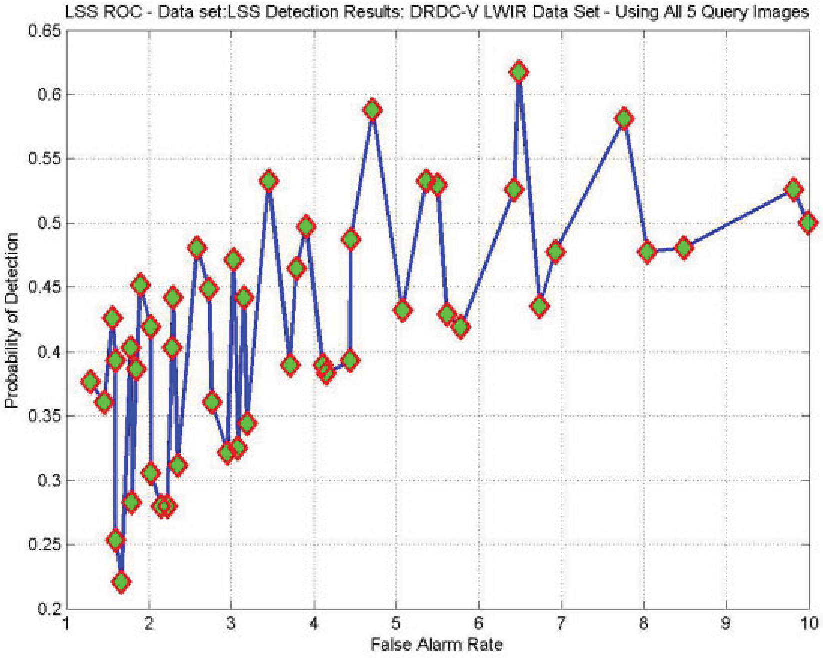


Figure 3-12 ROC Generated by Varying LSS Detection Parameters for DRDC-V LW-IR Data

Similar to the HOG SVM performance analysis above, the combination of input parameters yielding the best experimental detection results is used to illustrate the sensitivity of the probability of detection and false alarm rate to the variation of the input parameters according to the following two combined variations:

- Vary one parameter while fixing the other two parameters to their best results values
- Vary two parameters while fixing the third parameter to its best result value.

Figure 3-13, Figure 3-14 and Figure 3-15 illustrate the sensitivity of the false alarm rate and probability of detection to the variation of the window-stride, Hough-bin stride and score-threshold, respectively. In each case, the indicated LSS parameter is varied while the other two parameters are fixed to their respective values yielding the best detection results, as specified in Table 3-4. In view of these figures, we observe the following:

- The detection probability decreases while the false alarm rate increases as the window-stride increases. This suggests that an even smaller value of the window-stride, i.e. value of 1, may result in even better detection results. However, this would come at a significantly high computational complexity, as LSS descriptor for every pixel in the image will be computed and matched. As we shall discuss in Section 3.1.2.4, it takes about 17.5 seconds to process each image when the window stride parameter is set to 2. Lowering the value of this parameter to 1 is expected to at least quadruple the average processing time per frame.
- The score threshold is inversely proportional to the probability of detection and false alarm rate, as expected.
- As the Hough bin size increases, the probability of detection and false alarm rate decrease. However, as the Hough bin size increases, the position of the detected bounding boxes would become less accurate.

Figure 3-16, Figure 3-17 and Figure 3-18 illustrate the sensitivity of the false alarm rate and probability of detection to the simultaneous variation of two LSS detection parameters, while keeping the third parameter fixed to its value yielding the "best" detection results, as specified in Table 3-4.

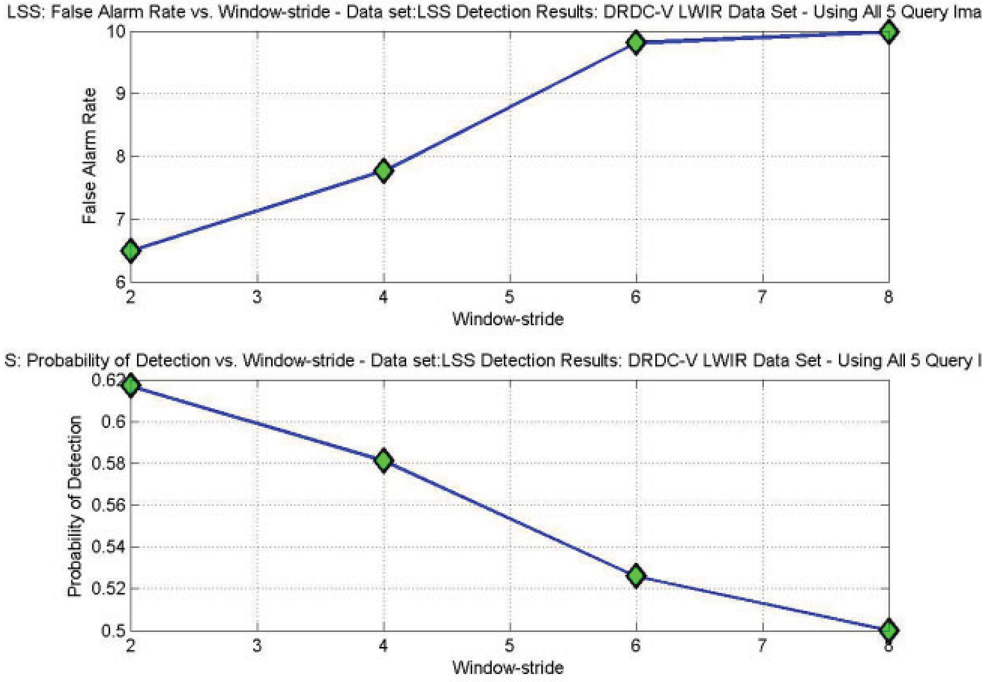


Figure 3-13 Sensitivity of Results to Variation of Window-stride Only for DRDC-V LW-IR Data

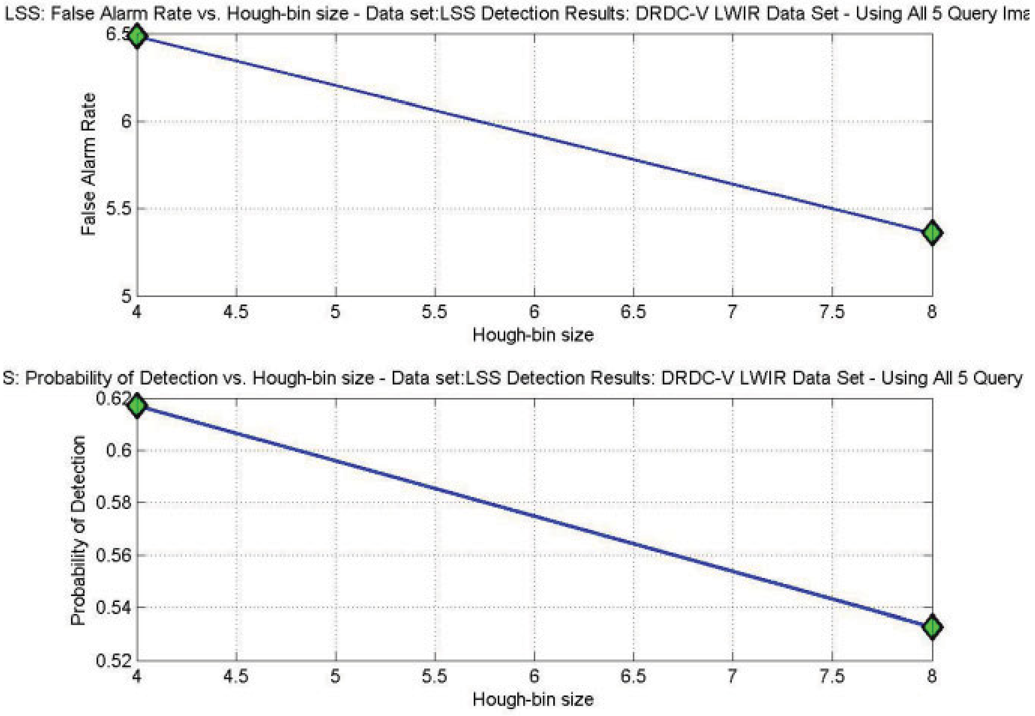


Figure 3-14 Sensitivity of Results to Variation of Hough-bin Size Only for DRDC-V LW-IR Data

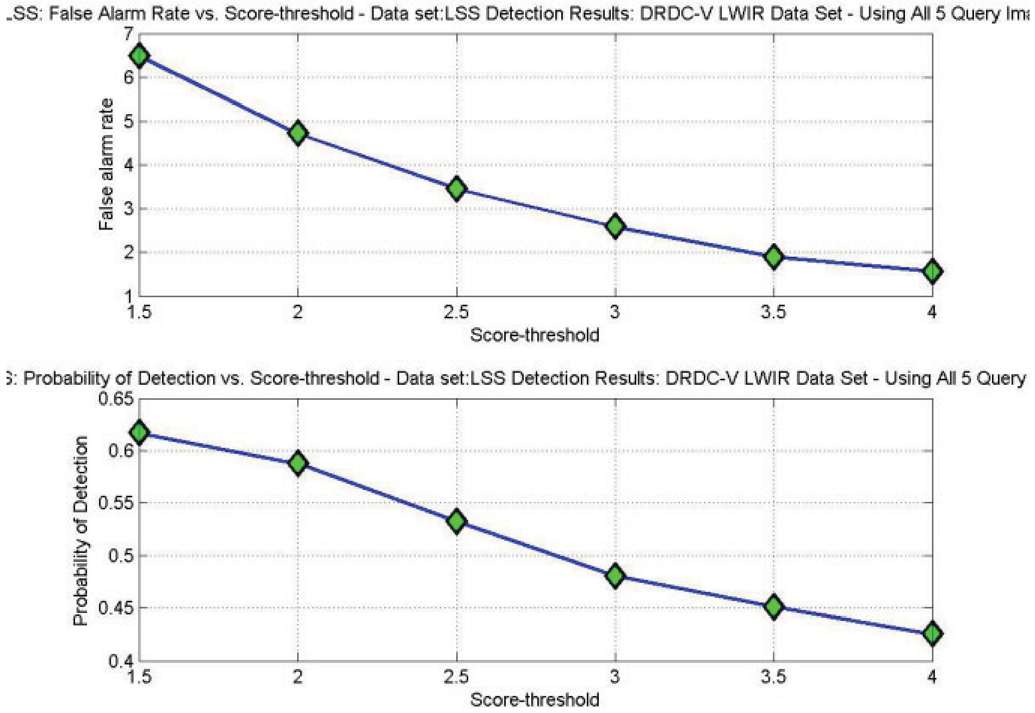


Figure 3-15 Sensitivity of Results to Variation of Score-Threshold Only for DRDC-V LW-IR Data

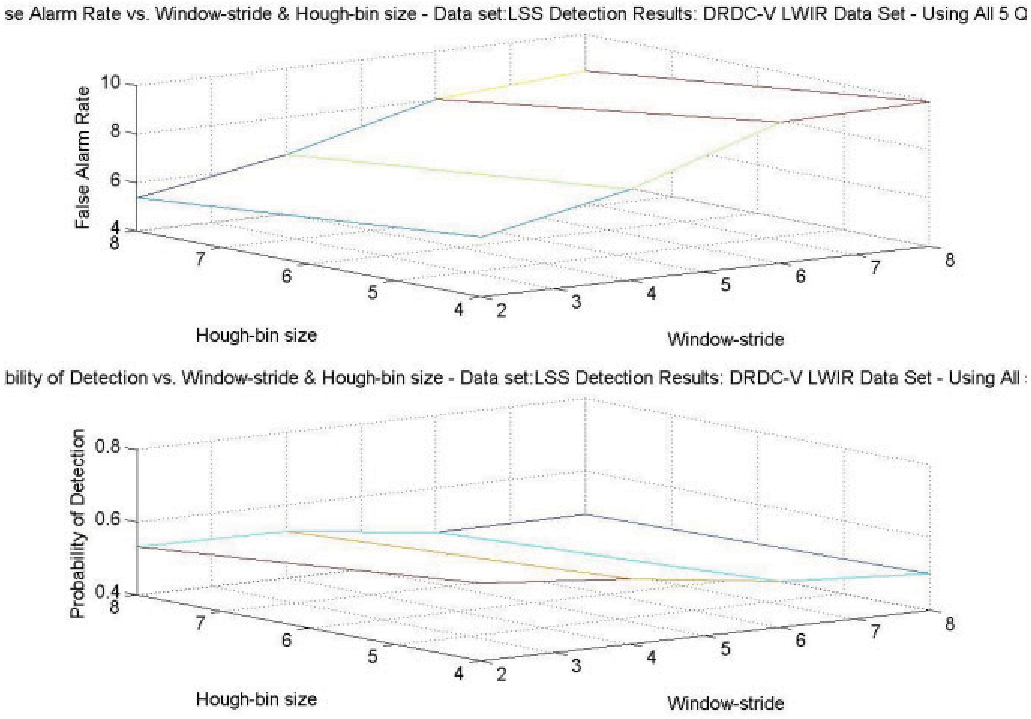


Figure 3-16 Effects of Variation of Window-stride and Hough-bin Size for DRDC-V LW-IR Data

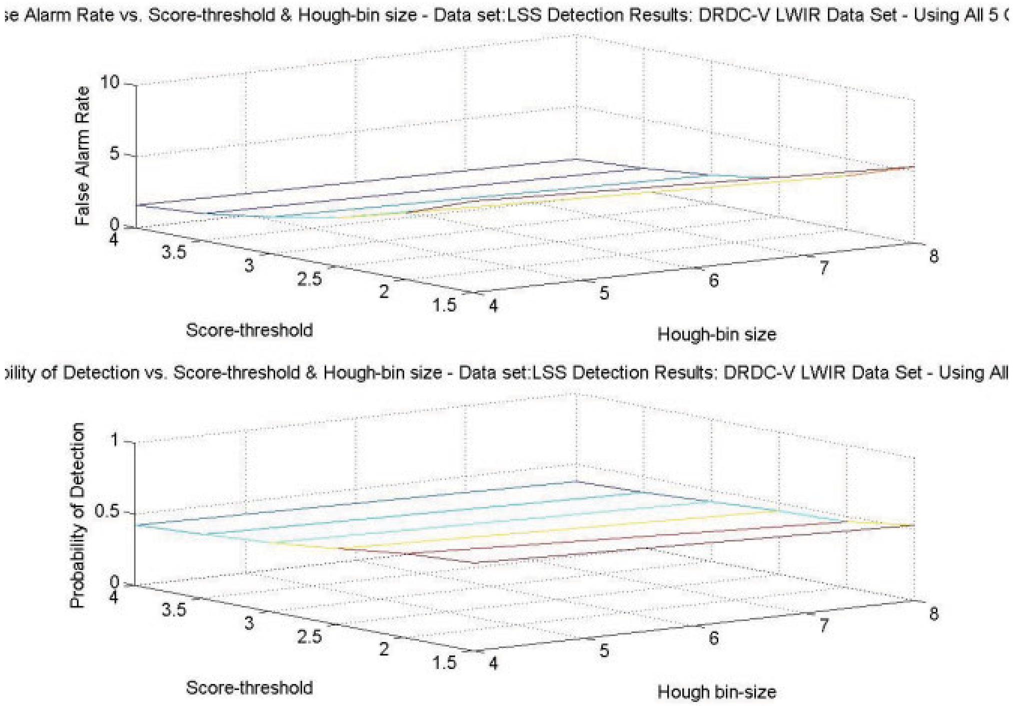


Figure 3-17 Effects of Variation of Hough-bin Size and Score Threshold for DRDC-V LW-IR Data

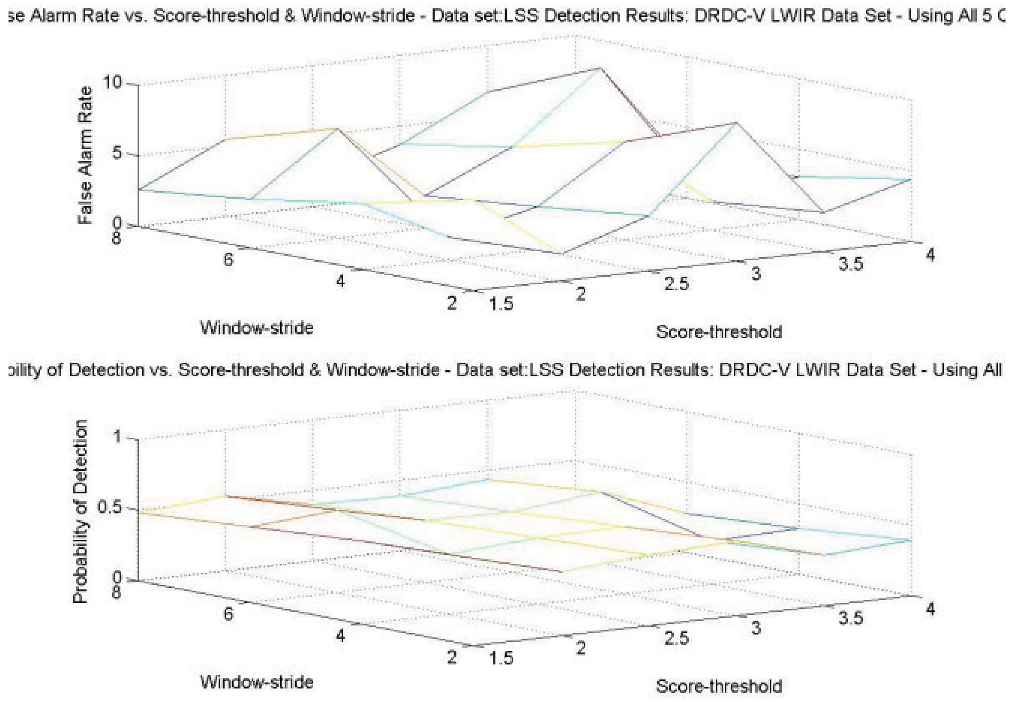


Figure 3-18 Effects of Variation of Window-stride and Score Threshold for DRDC-V LW-IR Data

3.1.2.4 Processing Time

Figure 3-19, Figure 3-20 and Figure 3-21 illustrate the dependence of the processing time on LSS detection parameters. We note that the processing time is highly sensitive to the value of the window-stride LSS parameter, resulting in an average of about 17.5 seconds per image on Intel Xeon W3520 2.67GHz CPU when this parameter is set to 2. Decreasing this parameter even further would at least quadruple the average processing time per frame, which is not practical for real-time processing applications.

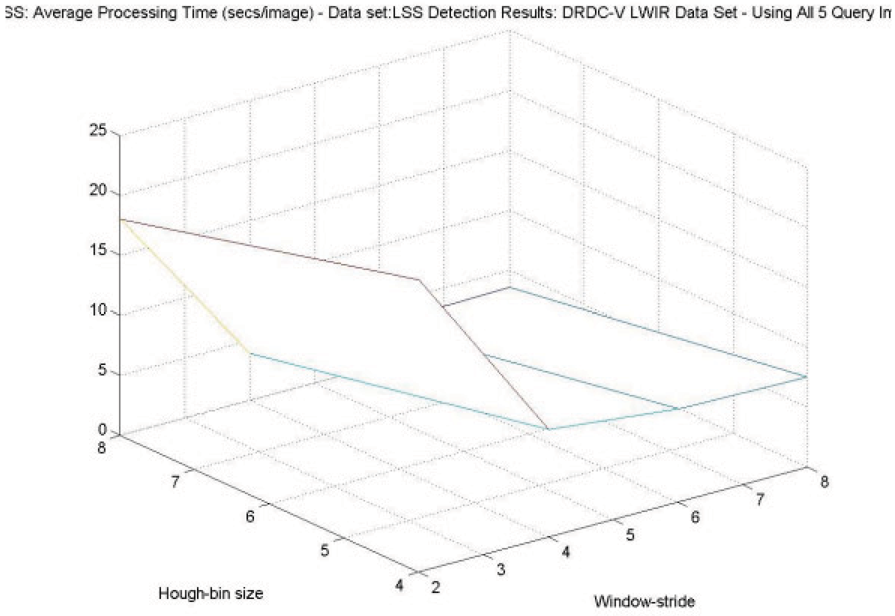


Figure 3-19 Processing Time Dependence on Variation of Hough-bin-size and Window-stride for DRDC-V LW-IR Data

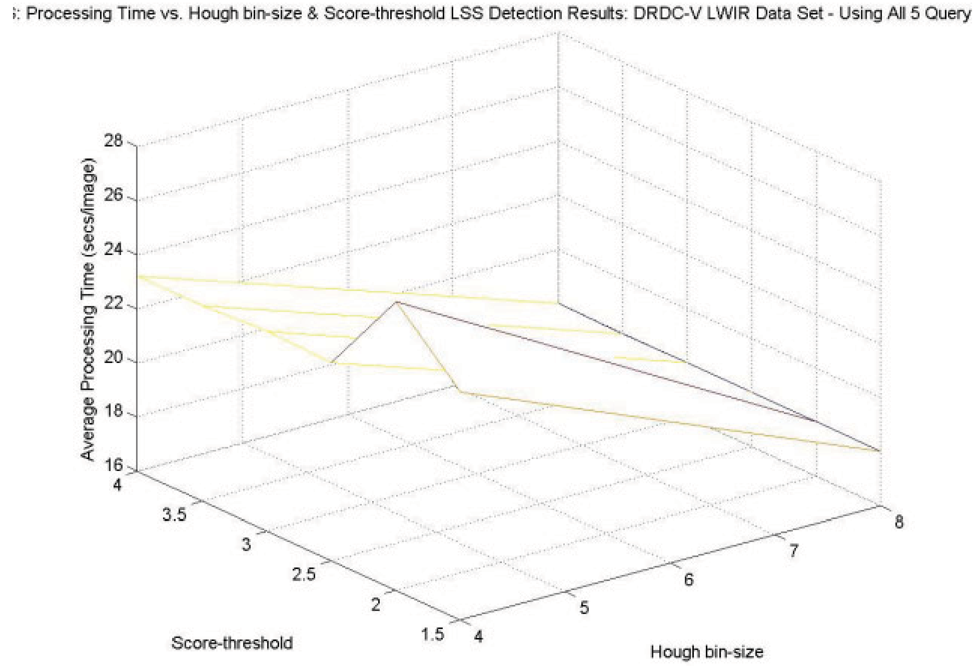


Figure 3-20 Processing Time Dependence on Variation of Hough-bin Size and Score Threshold for DRDC-V LW-IR Data

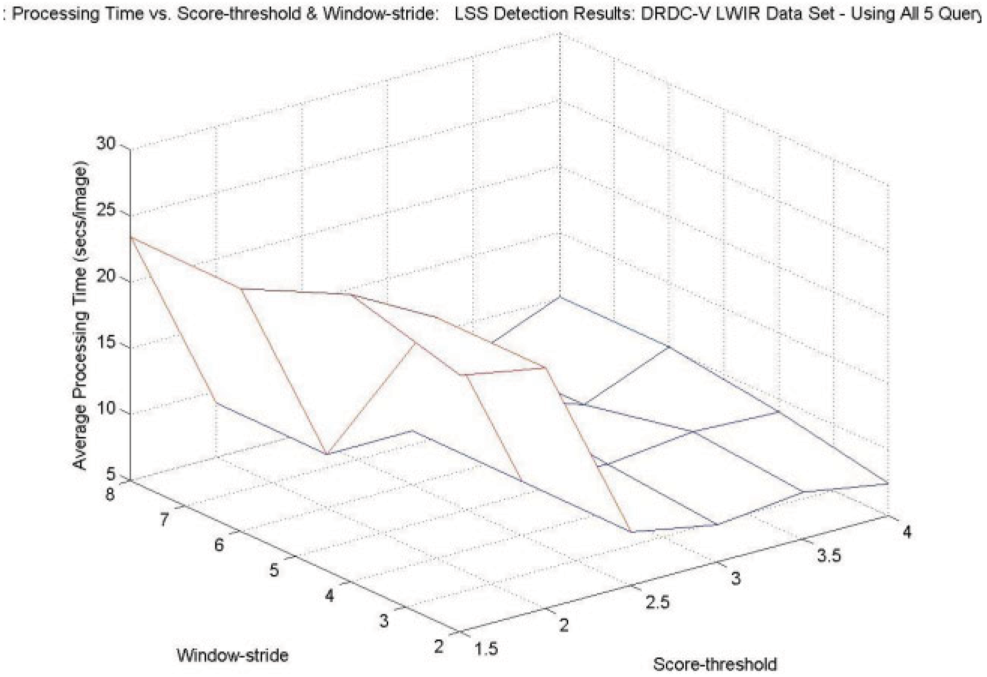


Figure 3-21 Processing Time Dependence on Variation of Window-stride and Score-threshold for DRDC-V LW-IR Data

3.1.2.5 Effects of the Number of Query Images

Figure 3-22 illustrates the LSS detection results using different number of query images. As expected, increasing the number of query images leads to an increase in both the probability of detection as well as the false alarm rate. In practice, it is expected that the LSS will be used in conjunction with a sufficiently large number of different templates, depicting distinctly appearing human beings. For example, the various query images should have different human-body poses, including walking, running, sitting, leaning, stretching, signalizing, etc. Moreover, the query images could also capture a variety of appearances of the humans' images, such as near and far range images, partially occluded human images, etc. This variability between the query images should increase the likelihood of detecting people with different appearances in the test imagery data.

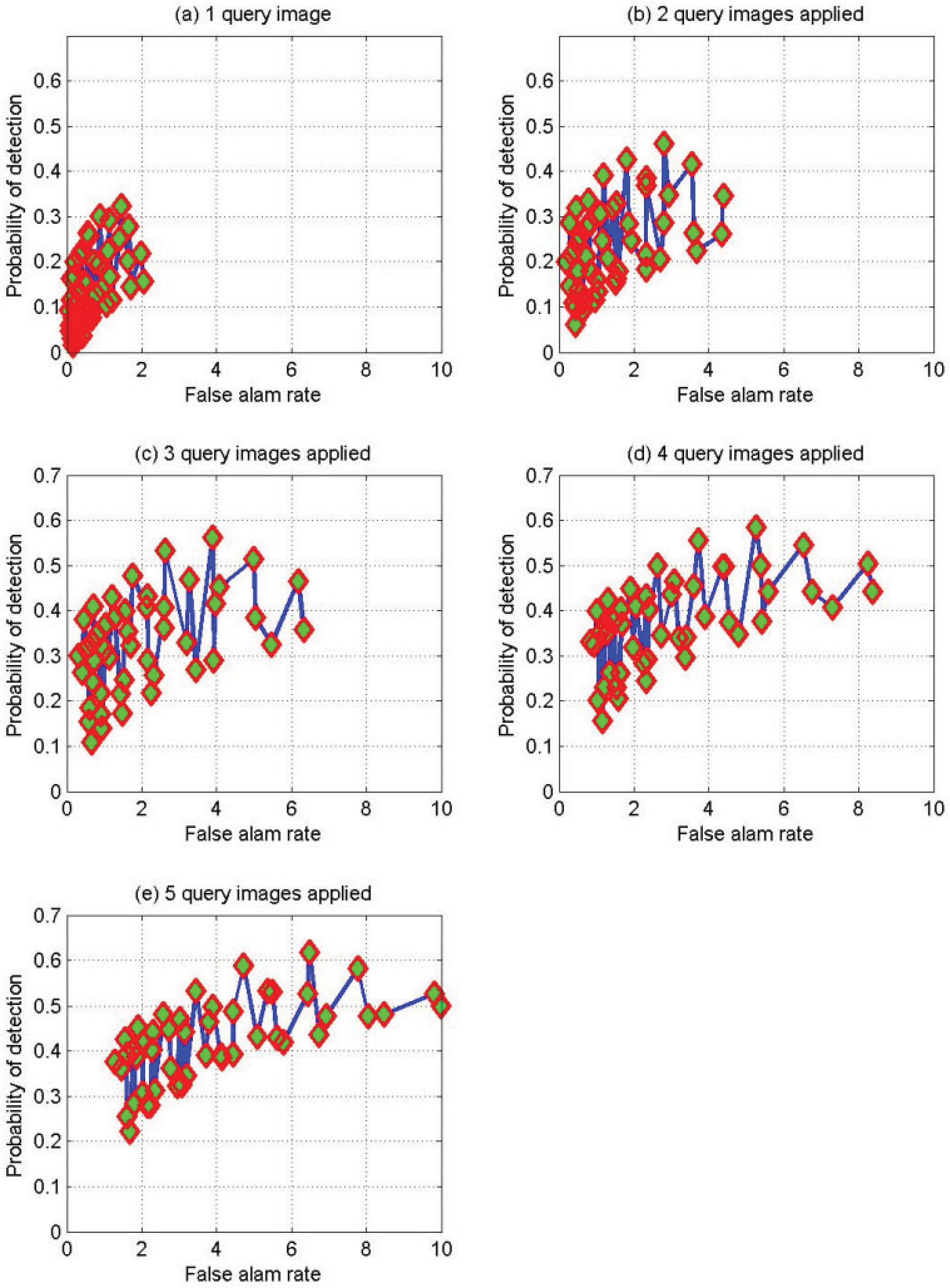


Figure 3-22 Effects of Query Images Selection on LSS Detection Results for DRDC-V LW-IR Data

3.2 UK Trial LW-IR Data

3.2.1 HOG SVM

3.2.1.1 SVM Training and Test Data

Table 3-5 illustrates how the available data is sub-divided into training and test subsets. As before, the SVM training is based on manually selected ground truth from the training subset of images. For each image in the training set, we select the following:

- Positive examples: All persons in the image
- Negative examples: Two examples are randomly selected that do not overlap with each other or with the positive examples.

In an effort to increase the positive and negative examples training set, we also flip each exemplar image, with respect to vertical central axis, resulting in twice as many positive and negative exemplars.

Table 3-5 Training and Test Data for UK Trial LW-IR Data

#	Training	Processing
1	4dec\d2	4dec\d1
2	5dec\d2	5dec\d1

3.2.1.2 SVM Re-Training

As mentioned above, initially the negative exemplars consist of two randomly selected chips, extracted from the training set in such a way that they do not overlap with each other or with the positive example images. However random selection of negative examples may not be the most effective way, as quite often most of these negative exemplars are easily distinguishable from an image of a human being. Hence, these negative exemplars do not represent image features, which are more likely to be confused with a person, such as vertical structures, including poles, bushes and small trees. In order to reduce the false alarm rate, the negative exemplars set should consist of a variety of non-human examples including images vastly different from human beings as well as images which may be confused for a human being.

Ideally, negative examples should be manually selected as part of the ground-truth annotation. That is, as one annotates all human beings in a frame, one would also annotate a few negative exemplars, which are not human being but may be confused with human beings. The manually selected negative examples may include vertical structures such as small trees, bushes and poles.

For each data set listed in Table 3-1, the provided ground truth includes only positive examples of human beings but no negative examples. In an effort to select more appropriate negative exemplars, we applied the following re-training process similar to [R-5]:

1. Use the positive and randomly selected negative ground-truth exemplars described in Section 3.2.1.1 to train the SVM algorithm on the training data described in Table 3-5 and generate the initial SVM model parameters.
2. Run the HOG SVM on the **training data set** described in Table 3-5 using the initial SVM model parameters and save the detected false alarms.
3. Add the detected false alarms to the set of randomly selected negative exemplars and use the expanded set of negative exemplars along with the set of positive exemplars to re-train the SVM algorithm using the same training data in order to generate new SVM model parameters.
4. Process the test data using the newly generated SVM model parameters, from re-training.

Next, we illustrate the HOG SVM detection results using the initial SVM model parameters as well as the newly generated SVM models parameters, after re-training.

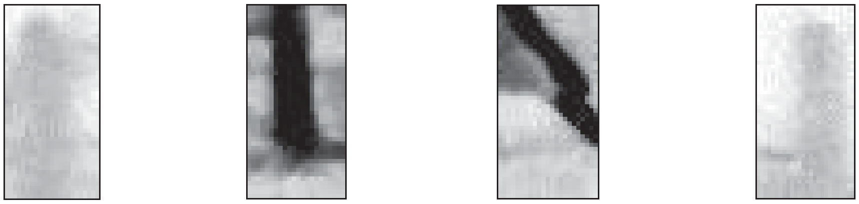
3.2.1.3 ROC Curves

Table 3-6 illustrates the ranges of variation for each test HOG SVM detection parameter and for the different data sets and training scenarios. As illustrated in the previous section, the scale HOG detection parameter was shown to have an "optimal" value of 1.05 and thus this parameter is fixed at this value.

Table 3-6 Variability Ranges of HOG SVM Detection Parameters for UK Trial LW-IR Data

Data sub-set	Data sub-set (Training)	HOG SVM Parameter	Minimum value	Step	Maximum value
1	04Dec2013 – d1 (Initial training)	Hit-threshold	7.0	0.25	12.0
		Final-threshold	8.0	0.5	12.0
	04Dec2013 – d1 (After re-training)	Hit-threshold	11.0	0.5	16.0
		Final-threshold	8.0	0.5	15.0
2	05Dec2013 – d1 (Initial training)	Hit-threshold	7.0	0.25	12.0
		Final-threshold	8.0	0.5	12.0
	05Dec2013 – d1 (After re-training)	Hit-threshold	11.0	0.5	16.0
		Final-threshold	8.0	0.5	15.0

Figure 3-24 illustrates the ROC curve resulting from varying the two HOG SVM key parameters, as specified in Table 3-6. It is evident that, for both test data subsets, the implemented SVM re-training strategy results in a significant improvement of the HOG SVM detection results. As discussed above, this is mainly due to expanding the set of negative exemplars, to include negative examples, which resemble a human being, in some shape or form. Figure 3-24 illustrates some of the negative exemplars, which were added to the set of negative examples used to re-train the SVM algorithm and generate the new SVM model parameters. These negative exemplars, which are false alarms detected by the HOG SVM algorithm based on the initial SVM training, depict vertical structures, which may be confused with human beings.



5dec_d2_image_29 5dec_d2_image_100 5dec_d2_image_101 5dec_d2_image_101

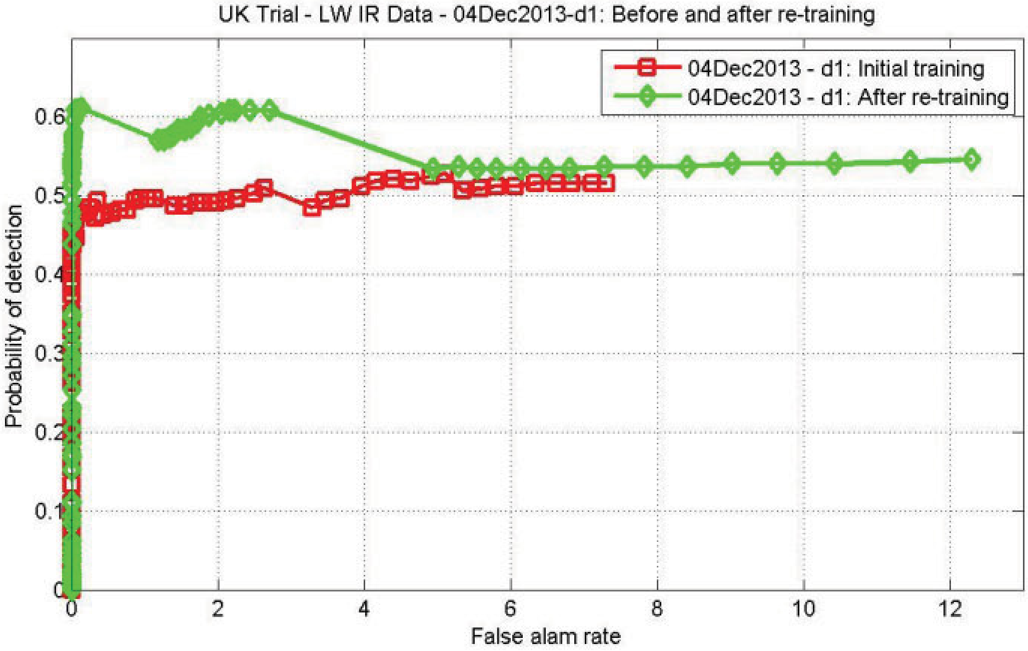
Figure 3-23 Sample HOG SVM False Alarms for UK Trial LW-IR Data

However, as we shall see for the UK Trial SW-IR data set, the SVM re-training strategy does not always result in an improvement of the detection results, as it did for this UK Trial LW-IR data set.

3.2.2 LSS Matching

3.2.2.1 Query Images

Figure 3-25 illustrates the LSS query images used for UK LW-IR data set. The directory and index of the image from which each query sub-image is extracted is also indicated. These templates were manually selected to capture different poses, sizes and shapes of human subjects. One may select a higher number of query images at the expense of high computational complexity and processing time. As discussed in Section 3.1.2.2, all 8 templates are supplied to the LSS simultaneously and the LSS combines their results in such a way that duplicate detections resulting from different query images are only counted one.



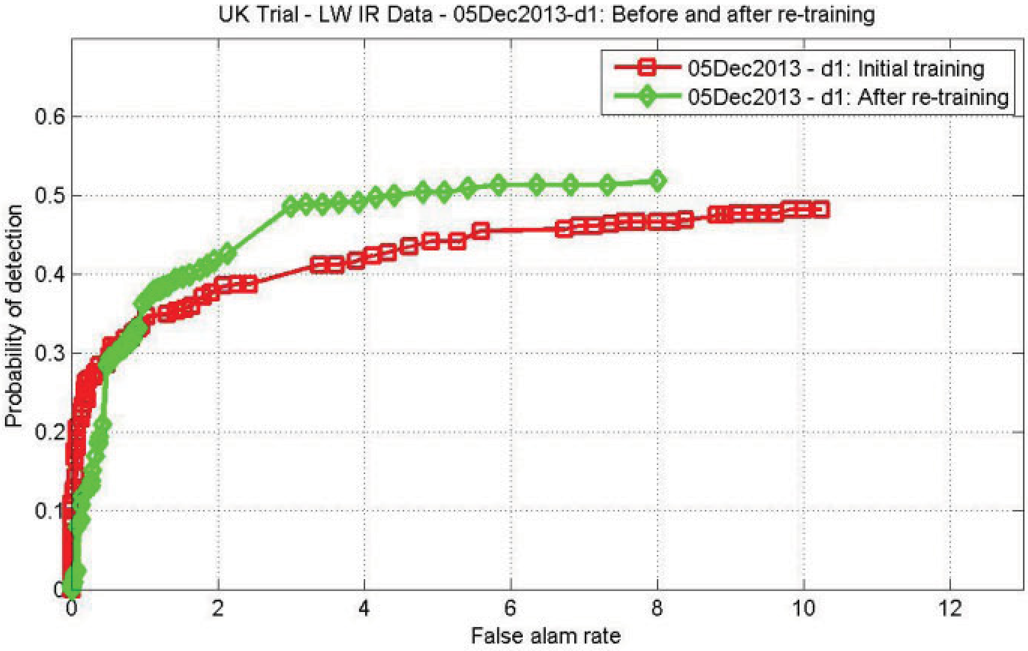
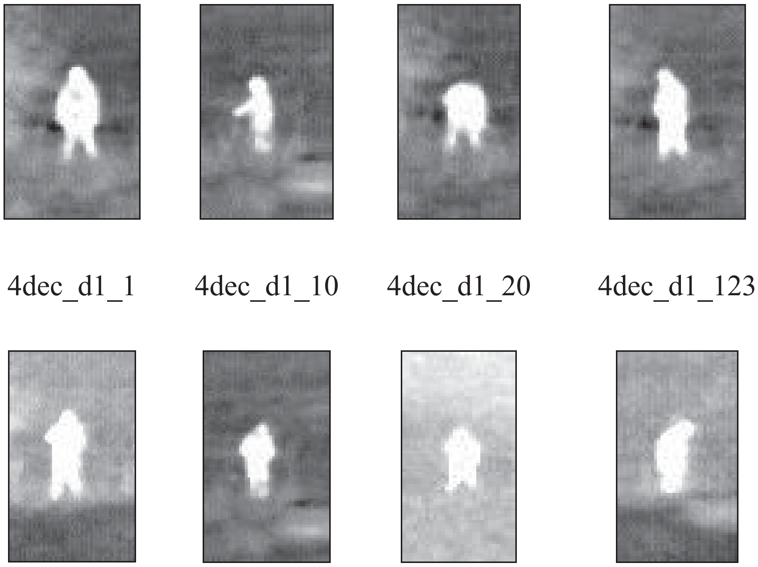


Figure 3-24 Effects of SVM Re-training on HOG Detection Results for UK Trial LW-IR Data



4dec_d1_1 4dec_d1_10 4dec_d1_20 4dec_d1_123

4dec_d2_125 4dec_d1_133 5dec_d1_147 4dec_d2_156

Figure 3-25 Selected Query Images for UK Trial LW-IR Data

3.2.2.2 ROC Curves

Figure 3-26 illustrates the ROC curve resulting from the varying the LSS key parameters, as specified in Table 3-7.

Table 3-7 Variability Ranges of LSS Detection Parameters for UK Trial LW-IR Data

LSS Parameter	Minimum Value	Step Size	Maximum Value
Window-stride	2	2	8
Hough bin-size	4	4	8
Score threshold	1.5	0.5	4.0

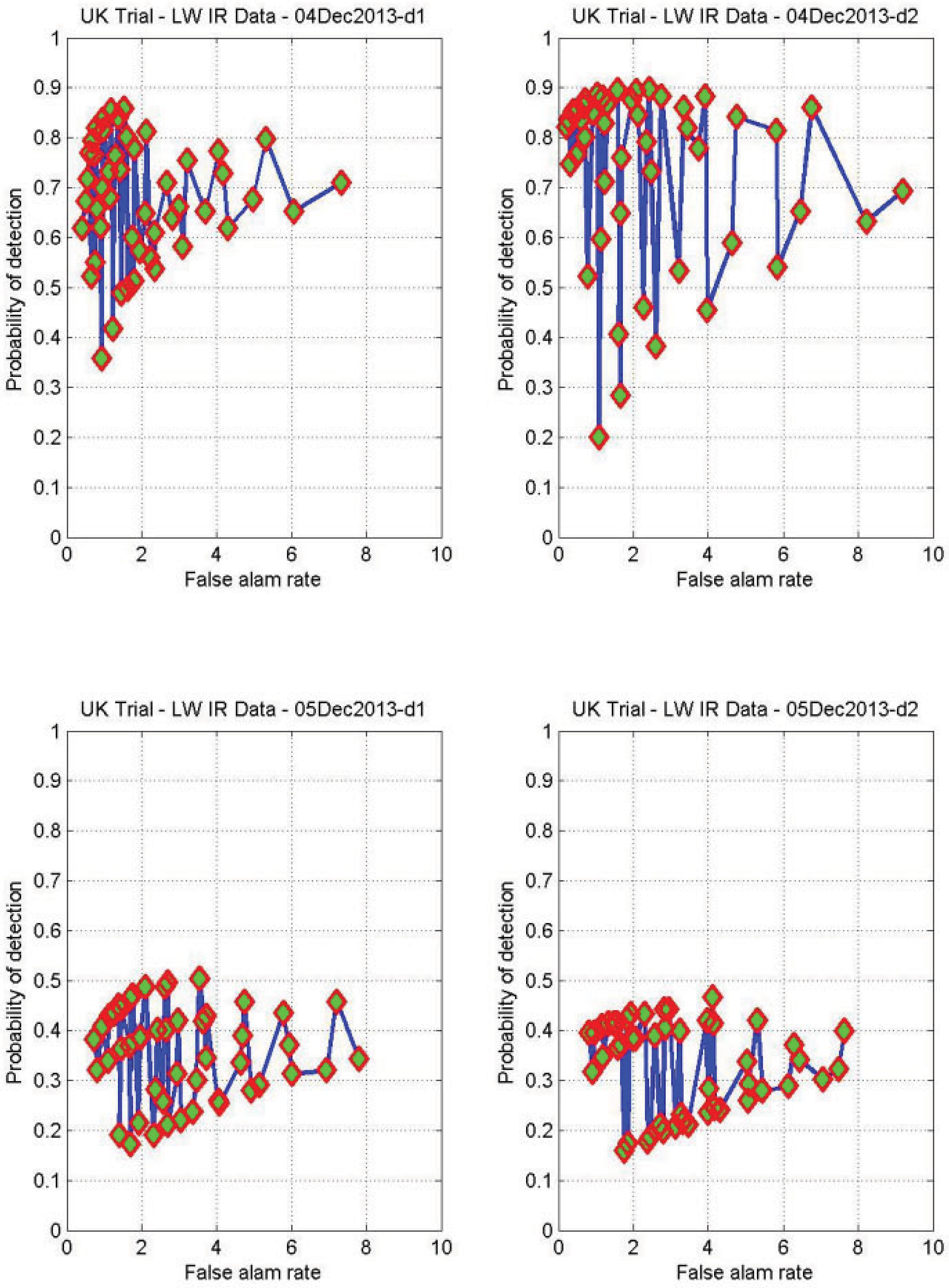
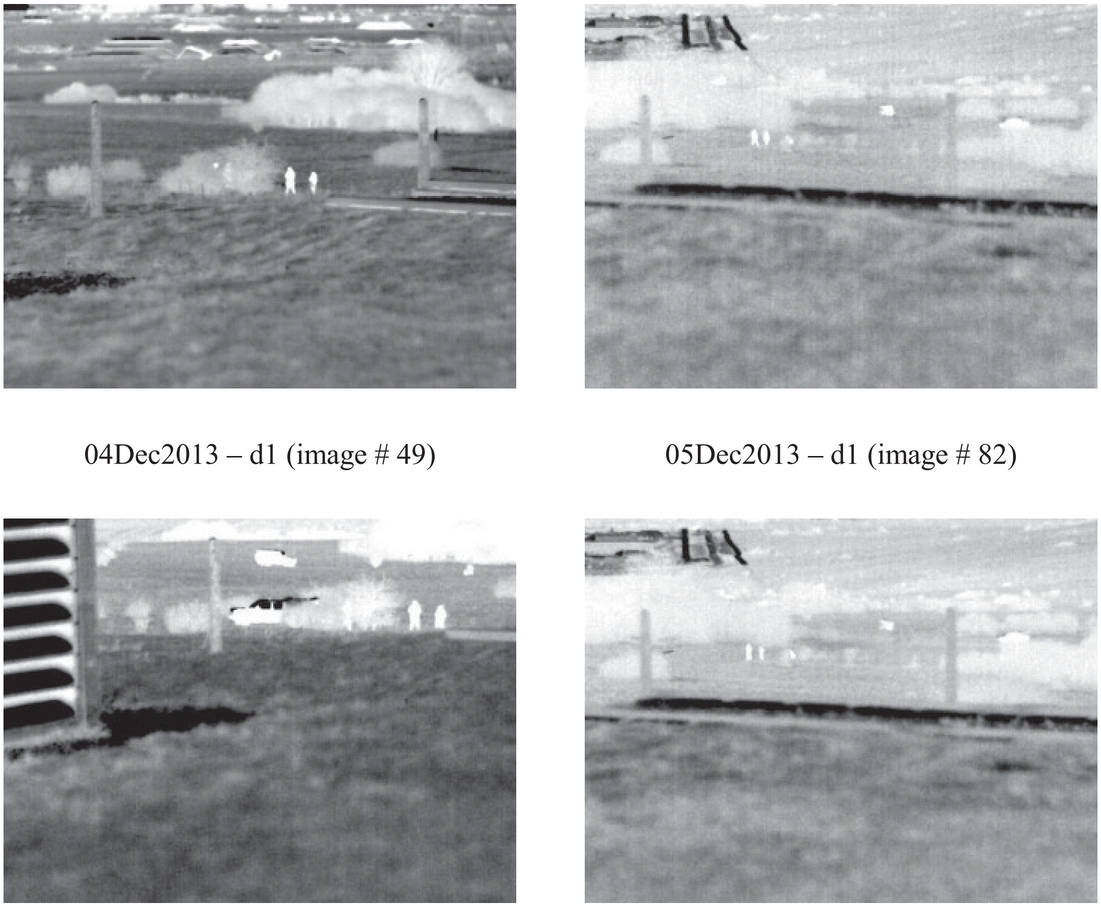


Figure 3-26 ROC Generated by Varying LSS Detection Parameters for UK Trial LW-IR Data

Clearly, the LSS matching results obtained for the 04Dec2013-d1 and 04Dec2013-d2 data subset are significantly better than those obtained for their 05Dec2013-d1 and 05Dec2013-d2 counter-parts. This is likely due to the significant differences in the image quality, contrast and noise-level due to gain, exposure, range and field of view, between the 04Dec2013 and 05Dec2013 data sets, as illustrated in Figure 3-27. These sample images indicate the higher image quality for the 04Dec2013 data sets as compared to their 05Dec2013 data sets.



04Dec2013 – d1 (image # 49)

05Dec2013 – d1 (image # 82)

04Dec2013 – d2 (image # 140)

05Dec2013 – d2 (image # 80)

Figure 3-27 Sample Images from 04Dec2013 and 05Dec2013 UK Trial LW-IR Data

3.3 UK Trial SW-IR Data

The SW-IR imagery is of 320×256 resolution, which is roughly a quarter of the resolution of the LW-IR imagery.

3.3.1 HOG SVM

3.3.1.1 SVM Training and Test Data

Table 3-8 illustrates how the available data is sub-divided into training and test subsets. As before, the SVM training is based on manually selected ground truth from the training subset of images. Negative exemplars are initially selected randomly from each training image, as described in Section 3.2.1.1.

Table 3-8 Training and Test Data for UK Trial SW-IR Data

#	Training	Processing
1	4dec\s4	4dec\s1
2	4dec\s5	4dec\s2
3	4dec\s6	4dec\s3
4	5dec\s2	5dec\s1

3.3.1.2 SVM Re-Training

As described in Section 3.2.1.2, the SVM classifier was re-trained using the expanded set of negative exemplars, which includes the initial randomly generated negative examples as well as hundreds of new negative exemplars. The additional negative exemplars are falsely detected by the HOG SVM based on the initial training of the SVM classifier.

3.3.1.3 ROC Curves

Table 3-9 illustrates the ranges of variation for the varied HOG SVM detection parameters using the different data sets and training scenarios. As before, the scale parameter is fixed at its experimental optimal value of 1.05.

Figure 3-28 illustrates the ROC curve resulting from the varying the HOG SVM key parameters, as specified in Table 3-9. Unlike the case for the UK Trial LW-IR data set discussed in Section 3.2.1.3, the SVM re-training does not appear to result in an improvement of the HOG detection results in this case. In fact, in some cases, the re-training appears to have degraded the HOG performance. We shall explore this issue further in Section 3.5 in order to understand the reasons behind this inconsistent performance of the SVM re-training.

Table 3-9 Variability Ranges of HOG SVM Detection Parameters for UK Trial SW-IR Data

Data sub-set	Data sub-set (Training)	HOG SVM Parameter	Minimum value	Step	Maximum value
1	04Dec2013 – s1 (Initial training)	Hit-threshold	8.0	0.5	16.0
		Final-threshold	8.0	0.5	12.0
	04Dec2013 – s1 (After re-training)	Hit-threshold	10.0	0.5	16.0
		Final-threshold	8.0	0.5	12.0
2	04Dec2013 – s2 (Initial training)	Hit-threshold	8.0	0.5	16.0
		Final-threshold	8.0	0.5	12.0
	04Dec2013 – s2 (After re-training)	Hit-threshold	10.0	0.5	16.0
		Final-threshold	8.0	0.5	12.0
3	04Dec2013 – s3 (Initial training)	Hit-threshold	8.0	0.5	16.0
		Final-threshold	8.0	0.5	12.0
	04Dec2013 – s3 (After re-training)	Hit-threshold	10.0	0.5	16.0
		Final-threshold	8.0	0.5	12.0
4	05Dec2013 – s1 (Initial training)	Hit-threshold	8.0	0.5	16.0
		Final-threshold	8.0	0.5	12.0
	05Dec2013 – s1 (After re-training)	Hit-threshold	10.0	0.5	16.0
		Final-threshold	8.0	0.5	12.0

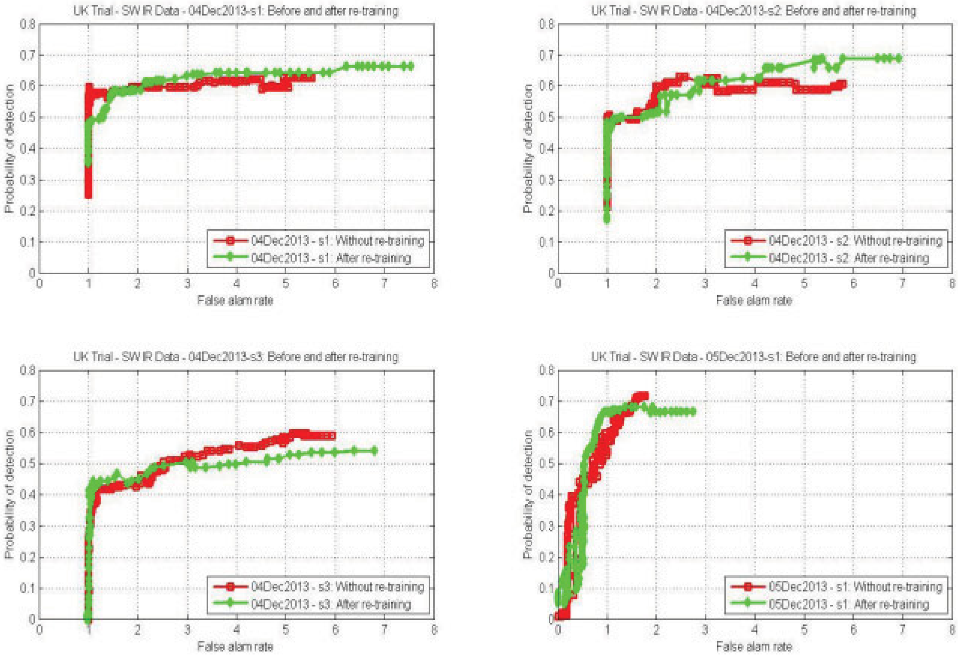


Figure 3-28 Effects of SVM Re-training on HOG Detection Results for UK Trial SW-IR Data

3.3.2 LSS Matching

3.3.2.1 Query Images

Figure 3-29 illustrates the LSS query images used for the UK Trial SW-IR data set. The directory and index of the image from which each query sub-image is extracted is also indicated. As before, these templates were manually selected to capture different poses, sizes and shapes of human subjects. Some of these query images depict soldiers carrying a weapon in different postures and poses. Again, one may select a higher number of query images at the expense of high computational complexity and processing time. As discussed in Section 3.1.2.2, all 10 templates are supplied to the LSS simultaneously and the LSS combines their results in such a way that duplicate detections resulting from different query images are only counted one.

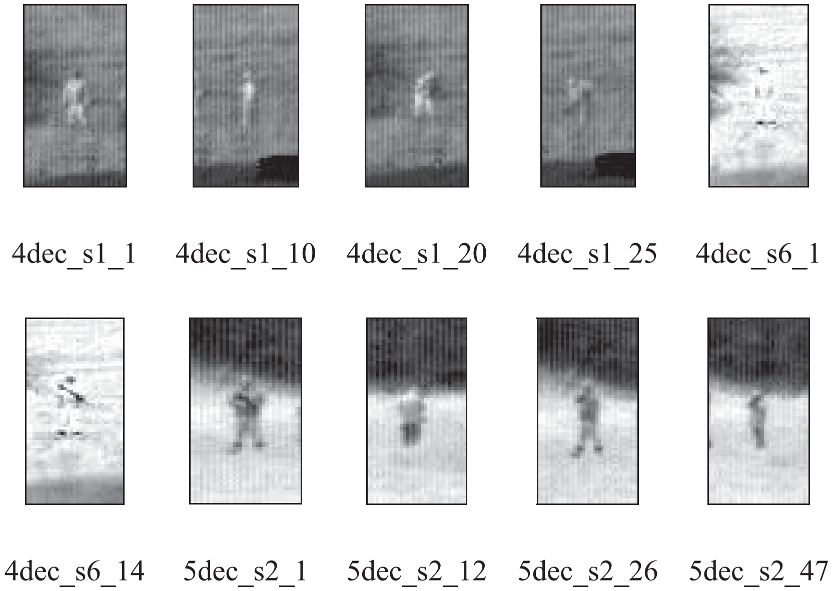


Figure 3-29 Selected Query Images for UK Trial SW-IR Data

3.3.2.2 ROC Curves

Figure 3-30 illustrates the ROC curve resulting from varying the LSS key parameters, as specified in Table 3-10. Clearly, the LSS appears to perform very well for all the UK Trial SW-IR data set, in some cases, achieving nearly 100% detection with less than one false alarm, despite its lower resolution than the LW-IR data set. This can be attributed to the good contrast and more details provided by SW-IR sensor for these data sets, as illustrated in Figure 3-31, and the scenes are more or less the same throughout the sequences.

Table 3-10 Variability Ranges of LSS Detection Parameters for UK Trial SW-IR Data

LSS Parameter	Minimum Value	Step Size	Maximum Value
Window-stride	2	2	8
Hough bin-size	4	4	8
Score threshold	1.5	0.5	4.0

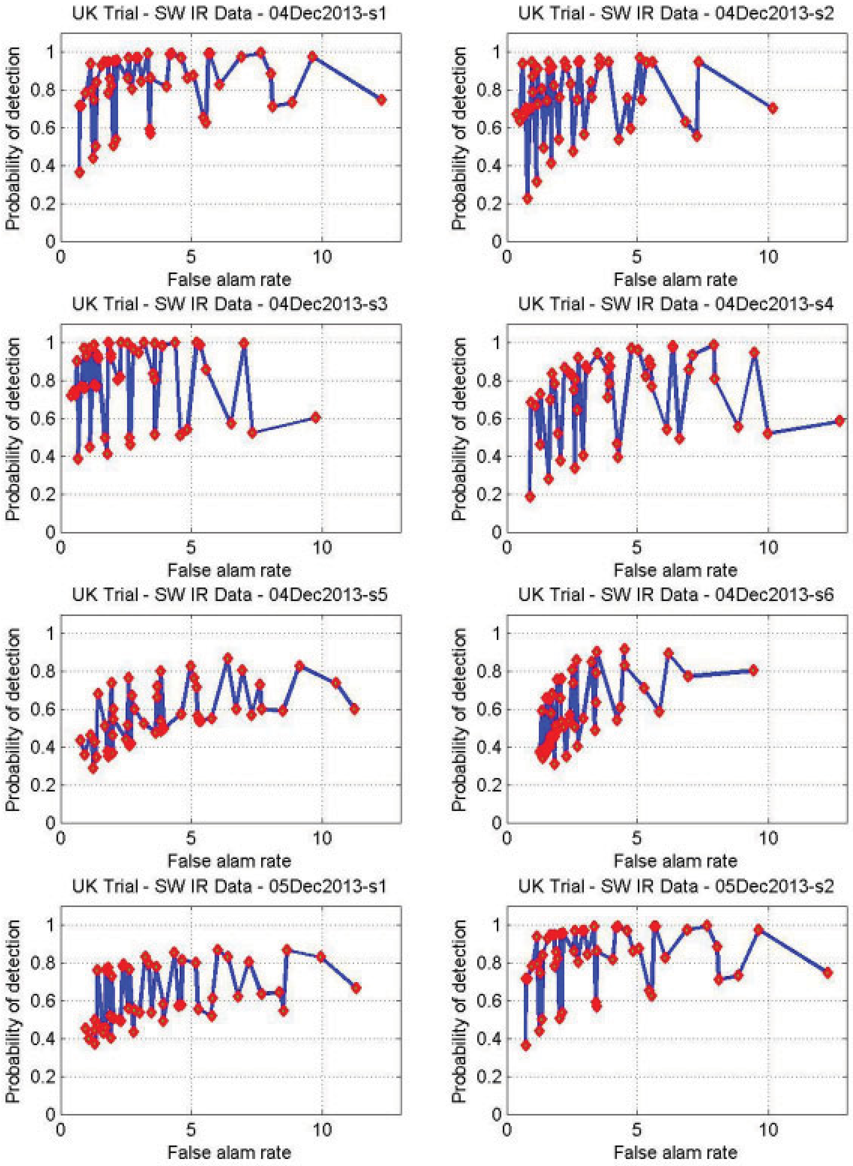
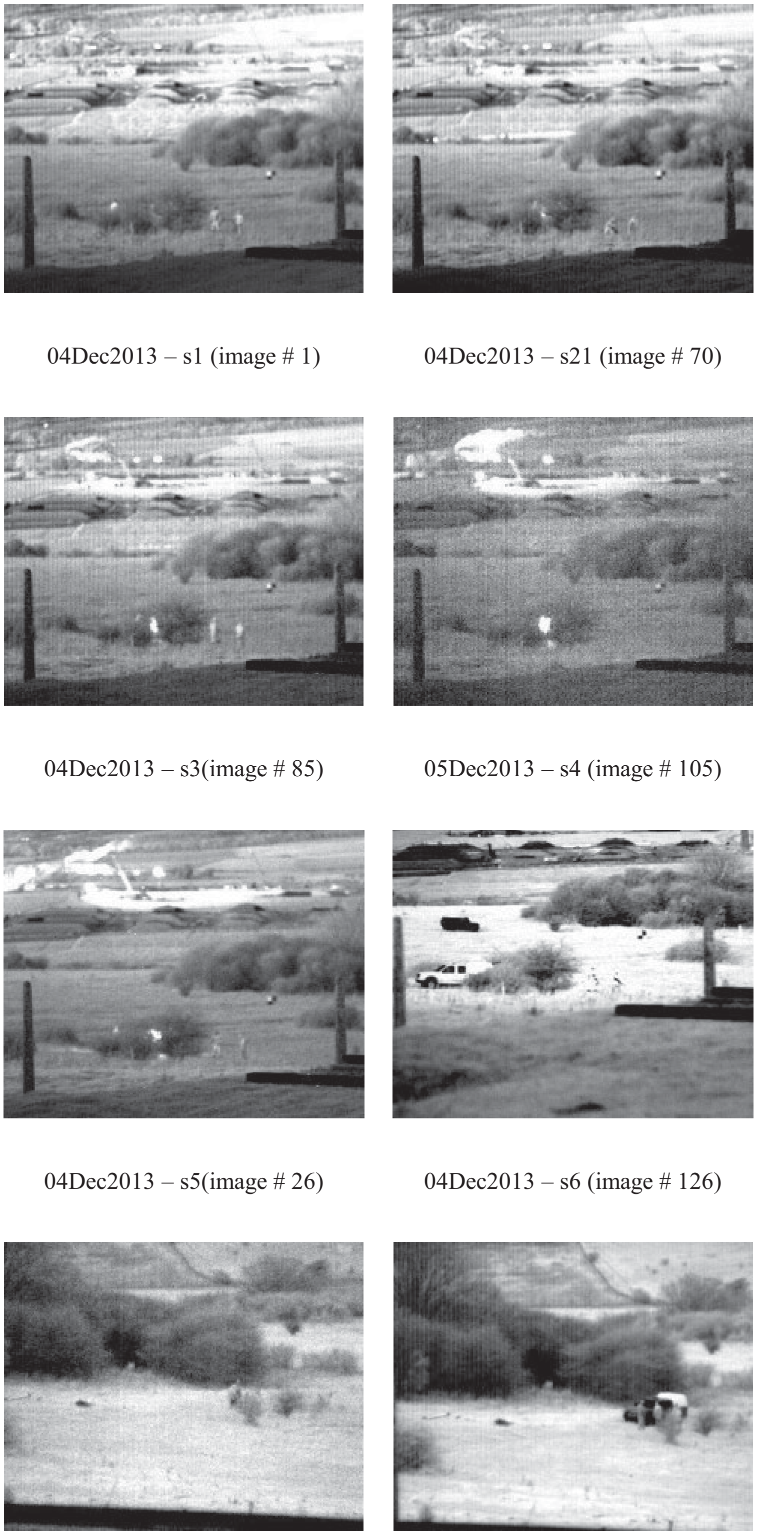


Figure 3-30 ROC Generated by Varying LSS Detection Parameters for the UK Trial SW-IR Data



04Dec2013 – s1 (image # 1)

04Dec2013 – s21 (image # 70)

04Dec2013 – s3(image # 85)

05Dec2013 – s4 (image # 105)

04Dec2013 – s5(image # 26)

04Dec2013 – s6 (image # 126)

05Dec2013 – s1 (image #33)

05Dec2013 – s2 (image # 66)

Figure 3-31 Sample Images from 04Dec2013 and 05Dec2013 UK Trial SW-IR Data

3.4 DRDC-V Visible Data

The DRDC-V visible imagery is of 1023×767 resolution, which is significantly higher than the LW-IR or the SW-IR data sets.

3.4.1 HOG SVM

3.4.1.1 SVM Training and Test Data

OpenCV includes SVM model parameters that were pre-trained for visible pedestrian images where the people are fairly large. A wide variety of positive and negative examples from tens of thousands of images are typically used for such training [R-5]. Since this visible data set resembles such training data, there is no need to train the SVM classifier ourselves. As it can be seen later, the HOG SVM detection indeed performs very well with the default OpenCV parameters. Therefore, we do not need to split this data set into training and testing subsets, and all the images can be used as test data.

3.4.1.2 ROC Curves

Table 3-11 illustrates the ranges of variation for the varied HOG SVM detection parameters using the different data sets and training scenarios. As before, the scale parameter is fixed at its experimental optimal value of 1.05.

Figure 3-32 illustrates the ROC curve resulting from the varying the HOG SVM key parameters, as specified in Table 3-11. Clearly, the HOG SVM performs very well for this data set. This may be attributed to the high image quality and the sufficiently large size of the human subjects in this data set, as depicted in Figure 3-33.

Table 3-11 Variability Ranges of HOG SVM Detection Parameters for DRDC-V Visible Data

HOG SVM Parameter	Minimum Value	Step Size	Maximum Value
Hit-threshold	0.0	0.25	5.0
Final-threshold	0.0	0.5	10.0

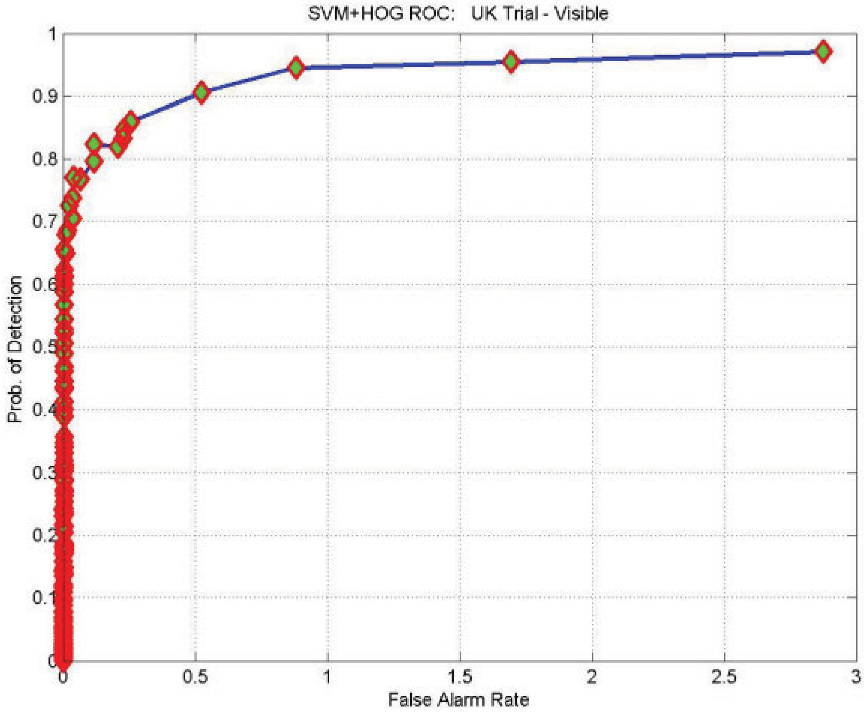


Figure 3-32 ROC Generated by Varying HOG SVM Detection Parameters for DRDC-V Visible Data

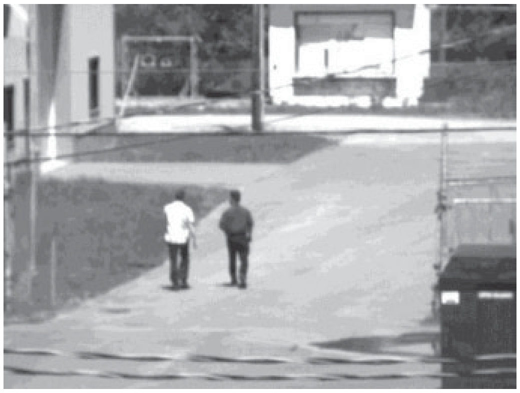


image # 25

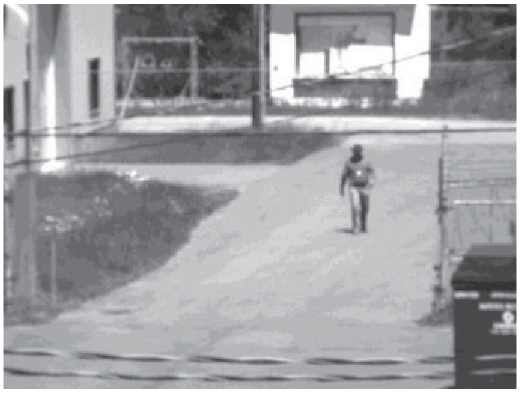


image # 73

Figure 3-33 Sample Images from DRDC-V Visible Data

3.4.2 LSS Matching

3.4.2.1 Query Images

Figure 3-34 illustrates the LSS query images used for the DRDC-V visible data set. As before, these templates were manually selected to capture different poses, sizes and shapes of human subjects. As discussed in Section 3.1.2.2, all 5 templates are supplied to the LSS simultaneously and the LSS combines their results in such a way that duplicate detections resulting from different query images are only counted one.

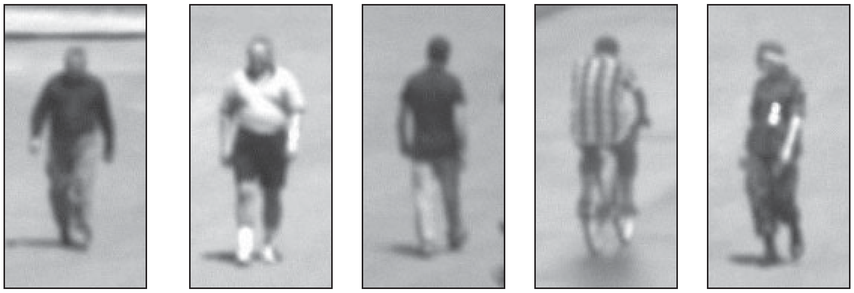


Figure 3-34 Selected Query Images for DRDC-V Visible Data

3.4.2.2 ROC Curves

Figure 3-35 illustrates the ROC curve resulting from varying the LSS key parameters, as specified in Table 3-12.

The LSS does not perform as well as the HOG SVM algorithm on this data. It is also significantly more computationally expensive than that the HOG SVM algorithm. The performance of the LSS algorithm may be improved by increasing the number of query images. Due to the high-resolution of the visible sensor, the query images are larger and more distinguishable with more details. Thus, in order to match the different appearances of human subjects, it may be necessary to use a higher number of query images.

Table 3-12 Variability Ranges of LSS Detection Parameters for DRDC-V Visible Data

LSS Parameter	Minimum Value	Step Size	Maximum Value
Window-stride	4	4	8
Hough bin-size	4	4	8
Score threshold	1.5	0.5	4.0

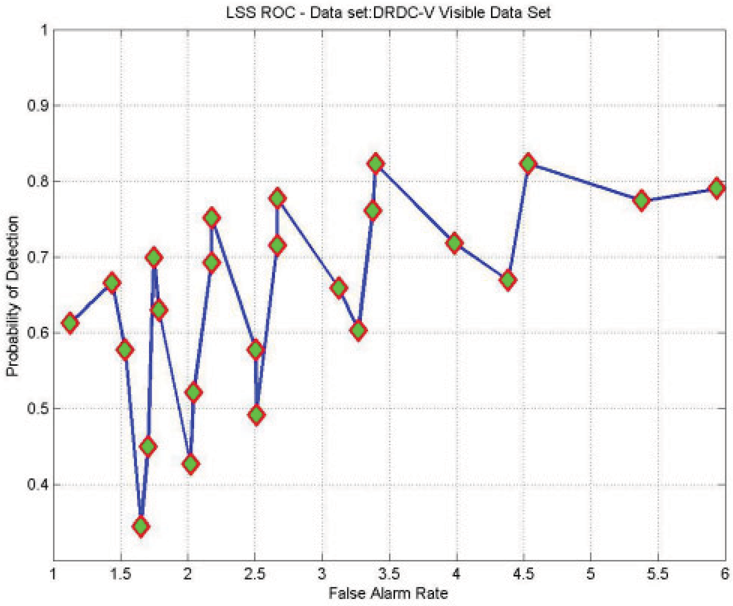


Figure 3-35 ROC Generated by Varying LSS Detection Parameters for DRDC-V Visible Data

3.5 Discussions

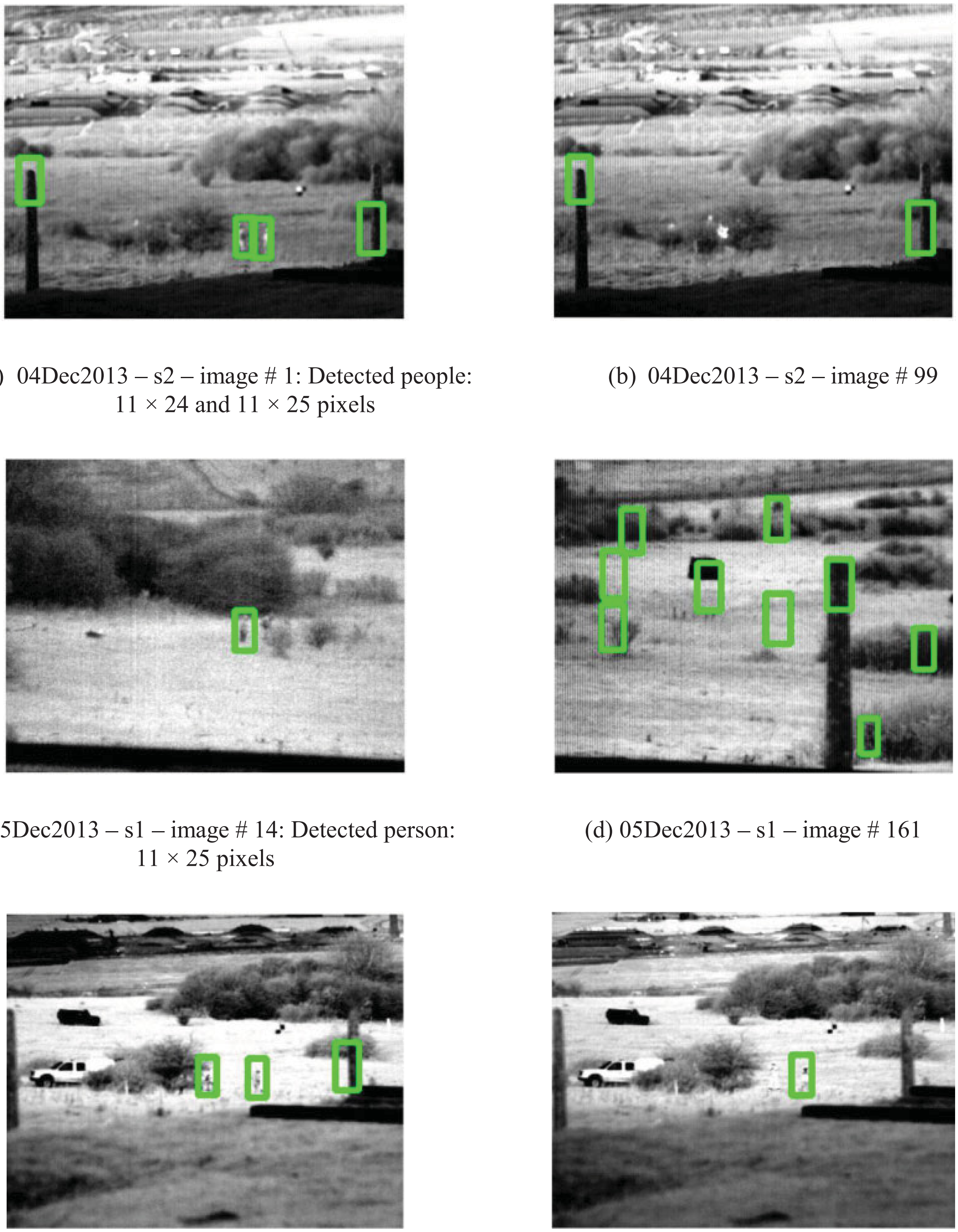
In view of the HOG SVM and LSS detection results for the various data sets presented, we make the following observations:

- There is high degree of variability in the HOG SVM and LSS detection results for the different data sets.
- Within each data set, there is also some degree of variability between the different sequences/folders of images, acquired on different dates. For example, as discussed in Section 3.2.2.2, there is significant difference between the LSS detection results for the `04Dec2013` and `05Dec2013` UK Trial LW-IR data subsets. These differences could be attributed to differences in the image quality of the two data sets due to changes in image parameters such as gain, exposure, range, field of view, etc.
- HOG SVM detection parameters:
    - A scale value of 1.05 seems to consistently yield the best detection results.
    - The optimal values of the other two threshold parameters, namely the hit and final thresholds, seem to vary from one data set to another.
- LSS matching parameters:
    - A window-stride value of 2 seems to consistently yield the best experimental detection results. However, this comes at high computational complexity and computing time of about 17 seconds per frame.

  - An even smaller value of the window-stride, i.e., a value of 1, may result in even better detection results. But this would come at a significantly high computational complexity, as LSS descriptors for every pixel in the image will be computed and matched. This is expected to at least quadruple the average processing time per frame, which would make it not practical for real-time processing applications.
  - The optimal values of the other two parameters, namely the Hough-bin size and score threshold, seem to vary from one data set to another.
- When processing the same data subset, most of the time, the best LSS matching results appear to be better than the best HOG SVM detection results. The main exception is the DRDC-V Visible data, where the HOG SVM yields better results than the LSS matching, using 5 query images.
- The HOG SVM algorithm (around 5 – 10 Hz) is significantly faster than the LSS matching algorithm.
- For the LSS algorithm, using more query images results in better probability of detection but at the expense of more false alarms and higher computational complexity.
- Re-training the SVM classifier does not consistently yield better HOG SVM detection results. This issue is further explored in Section 3.5.2.

3.5.1 Sample Detection Results

Figure 3-36 illustrates some sample HOG SVM detections, including true positives and false alarms. As noted previously, some of the false alarms correspond to vertical structures, such as trunks of trees, bushes and poles, which resemble the human subjects. The detected human subjects are relatively small size and appear to be at far range. In these sample results, the detected human subjects' images have width of 11 – 12 pixels and height of 19 – 25 pixels. We do not have sufficient information to estimate the range of the detected human subjects.



(a) 04Dec2013 – s2 – image # 1: Detected people: 11 × 24 and 11 × 25 pixels

(b) 04Dec2013 – s2 – image # 99

(c) 05Dec2013 – s1 – image # 14: Detected person: 11 × 25 pixels

(d) 05Dec2013 – s1 – image # 161

(e) 04Dec2013 – s6 – image # 148: Detected people: 12 × 19 pixels and 11 × 21.

(f) 04Dec2013 – s6 – image # 46: Detected person: 12 × 22 pixels

Figure 3-36 Sample HOG SVM Detection Results for UK Trial SW-IR Data

3.5.2 SVM Training

In this section, we examine the critical issue of the SVM training. As discussed in Sections 3.2.1.1, 3.2.1.2 and 3.3.1.2, we explored different approaches of training the SVM classifier using the UK Trial LW-IR and SW-IR data sets. The main difference between the training strategies lies in the selection of the negative exemplars. Initially, two negative exemplars were randomly selected from the input images, in such a way that they do not overlap with one another or the positive examples in the images. These randomly selected images were also flipped vertically to double the number of negative exemplars. We have found that in general, this yields good SVM models parameters resulting in good HOG detection results.

As detailed in Section 3.2.1.2, in an effort to improve the negative exemplars selection, we augmented the randomly generated set of negative exemplars with a generated set of false alarms and re-trained the SVM classifier. We found that HOG SVM algorithm yields mixed results after re-training. For the UK Trial LW-IR data sets, re-training the SVM classifier yields better HOG detection results, as illustrated in Section 3.2.1.2. However, for the UK Trial SW-IR data, re-training does not improve and sometimes degrades the HOG detection results, as illustrated in Section 3.3.1.2.

To gain a better understanding of this inconsistency in the effects of the SVM re-training, we take a closer look at the SVM classifier. As illustrated in Figure 3-37, a simplifier 2-dimensional binary SVM classifier attempts to discriminate between the computed features of two different classes. In our application, these two classes are “person” and “other”. When the SVM algorithm is properly trained using sufficiently comprehensive sets of positive and negative examples of the class of interest, the features of the two classes are sufficiently separable with a clear boundary between the two features sub-spaces. As such, the SVM classifier is expected to perform well, on average. The reader is referred to [R-8] for more details on the implementation and training of the SVM algorithm.

Figure 3-38 illustrates the case when the SVM classifier is trained using:

- Good positive exemplars of humans, based on ground-truth
- Randomly selected negative exemplars, as we initially trained our SVM classifier.

Since the negative exemplars are randomly selected, they are expected to have HOG features which are significantly separable from those of the positive exemplars. On one hand, this is good to have, as it makes the two classes quite separable. One the other hand, this is misleading because the randomly selected negative exemplars may not include many structures that resemble human subjects but are actually not human. This may include vertical structures such as tree trunks, poles and bushes, as illustrated in Figure 3-36. The reason for false alarms is because their HOG features are closer to those computed from the positive human exemplars than to those computed from the randomly selected negative exemplars, as illustrated in Figure 3-38.

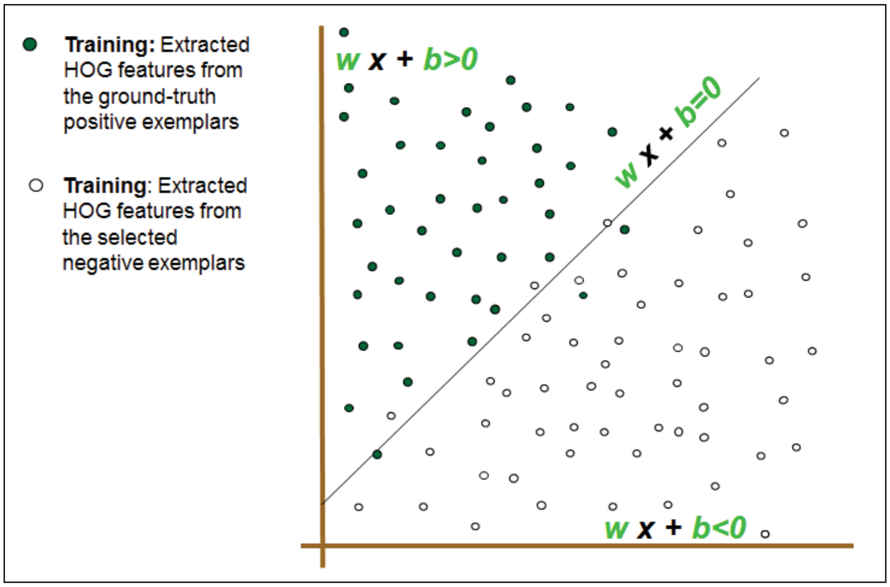

● Training: Extracted HOG features from the ground-truth positive exemplars
○ Training: Extracted HOG features from the selected negative exemplars
w x + b>0
w x + b=0
w x + b<0

Figure 3-37 SVM Classifier using Good Training Exemplars

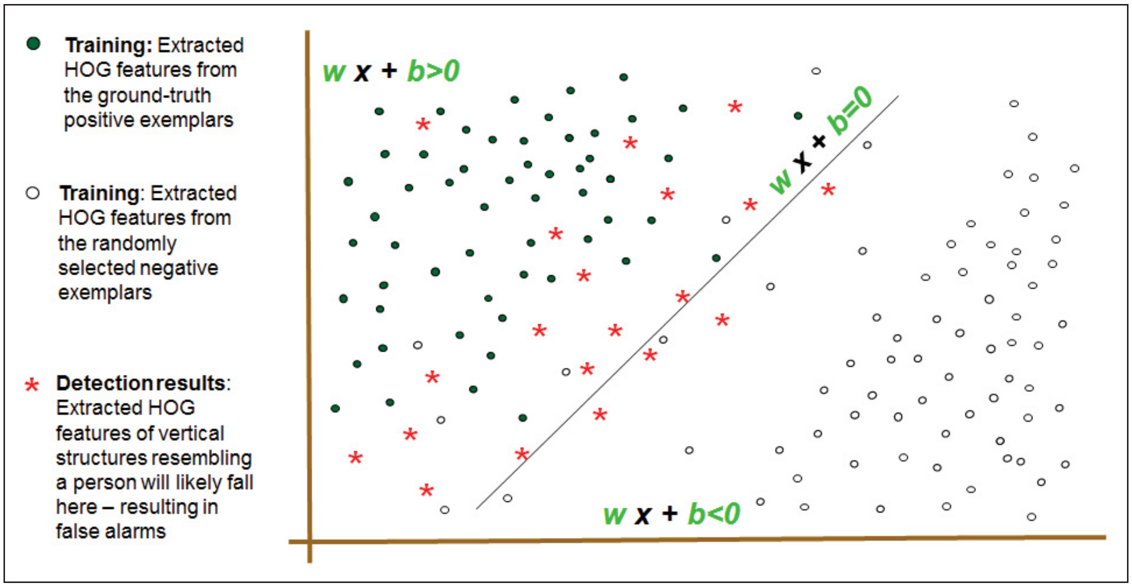

● Training: Extracted HOG features from the ground-truth positive exemplars
○ Training: Extracted HOG features from the randomly selected negative exemplars
* Detection results: Extracted HOG features of vertical structures resembling a person will likely fall here – resulting in false alarms
w x + b>0
w x + b=0
w x + b<0

Figure 3-38 SVM Classifier using Randomly Selected Negative Exemplars

Figure 3-39 illustrates what happens when one includes too many negative exemplars, which resemble human subjects. In this case, the HOG features of negative and positive examples are much closer together and the two classes become less separable, resulting is misclassifications. This may explain why re-training the SVM classifier sometimes degraded the HOG detection results for the UK Trial SW-IR data, as illustrated in Section 3.3.1.2.

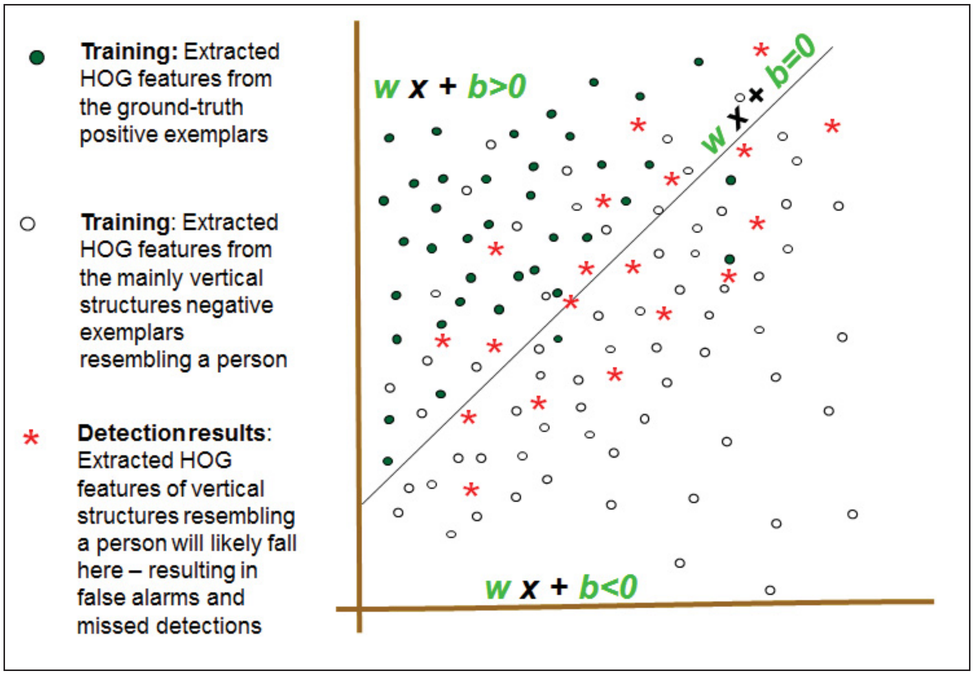

● Training: Extracted HOG features from the ground-truth positive exemplars
○ Training: Extracted HOG features from the mainly vertical structures negative exemplars resembling a person
* Detection results: Extracted HOG features of vertical structures resembling a person will likely fall here – resulting in false alarms and missed detections
w x + b>0
w x + b=0
w x + b<0

Figure 3-39 SVM Classifier including False Alarms as Training Negative Exemplars

In order to explore the case illustrated in Figure 3-39 even further, we have modified the set of negative exemplars as follows:

- Start with a set of false alarms detected by the HOG SVM algorithm based on an initial SVM training (using randomly generated negative exemplars)
- Examine and manually delete false alarms, except those resembling human subjects.

The new set of negative exemplars, consisting mainly of “person-resembling” images is then used along with the positive exemplars to re-train the SVM classifier. Figure 3-40 illustrated the HOG SVM detection results based on the re-trained SVM classifier of the UK Trial SW-IR data. We note that using the newly re-trained SVM classifier has degraded the HOG detection results even further. As explained above and illustrated in Figure 3-39, this may be due to the fact the negative exemplars may be too similar to the set of positive exemplars, and hence the two training sets become less separable in the HOG feature space.

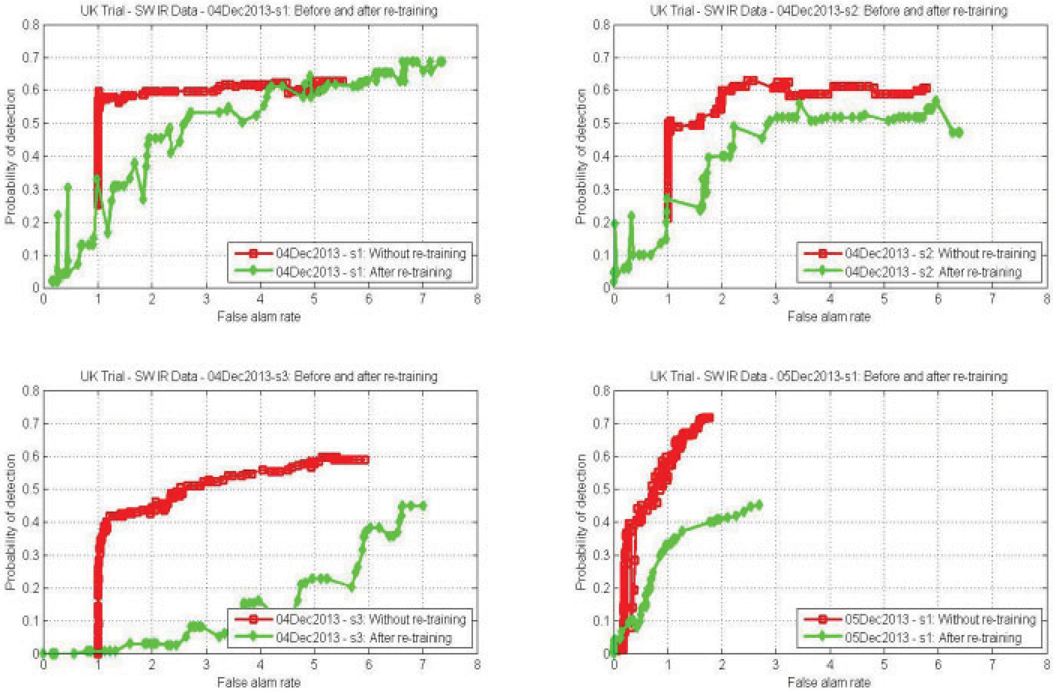


Figure 3-40 ROC Generated by Varying HOG SVM Detection Parameters Before and After Modified SVM Re-training for UK Trial SW-IR Data

Due to the inconsistencies in the performance of the SVM re-training strategy, it seems that initial SVM training based on randomly selected negative exemplars may be the better and more consistent approach. However, such a random selection of the negative exemplars has its limitations as discussed above and illustrated in Figure 3-38. To generate a more suitable set of negative exemplars, we suggest the followings:

- Manually select negative exemplars: As one is conducting grounding truth to annotate the actual human subjects in each image, one may also select a few negative examples, some consisting of vertical structures resembling a person and others more distinguishable from human.
- Most of the data sets in Table 3-1 appear to be acquired by a stationary camera resulting in more or less the same scene throughout the sequence. Image sequence captured by a moving camera is preferred, as this will allow for a wide variety of scenes, with human subjects in various backgrounds. A large, comprehensive and heterogeneous set of negative exemplars can be selected, which is more suitable for SVM training and is expected to improve the HOG SVM detection performance.

3.5.3 Recommended Processing Parameters

In practice, one can only tolerate a certain number of false alarms per frame. Thus, given a maximum false alarm rate the user is willing to tolerate, one can use the generated ROC curves to determine the highest probability of detection. Table 3-13 and Table 3-14 illustrate the best detection results, for all the processed data sub-sets, achieved by the HOG SVM and LSS matching algorithms respectively, while restricting the false alarm rate to be no more than 3 false detections per frame. A different table would be generated if the user can tolerate a different number of false alarms per frame.

Table 3-13 HOG SVM Detection Parameters for Various Data Sets to Yield False Alarm Rate ≤ 3

#	Data set	Sensor	Acquisition date	Folder	Hit-threshold	Scale	Final-threshold	Prob. of detection	False alarm rate	Rank
1	DRDC-V trial	LW-IR	NA	DRDC_IR_ Data	0.0	1.05	0.0	0.620	2.792	4
2	UK trial	LW-IR	04Dec2013 (after SVM re-training)	d1	16.0	1.05	8.0	0.552	**0.613**	6
				d2	Used for training SVM – Not processed					
			05Dec2013 (after SVM re-training)	d1	15.5	1.05	8.0	0.482	2.861	8
				d2	Used for training SVM – Not processed					
3		SW-IR	04Dec2013	s1	10.0	1.05	9.0	0.597	2.958	5
				s2	11.0	1.05	8.0	0.629	2.588	3
				s3	10.0	1.05	8.5	0.521	2.947	7
				s4	Used for training SVM – Not processed					
				s5						
				s6						
			05Dec2013	s1	8.0	1.05	8.0	0.718	1.778	2
				s2	Used for training SVM – Not processed					
4	DRDC-V trial	Visible	NA	DRDC_Vis_ Data	0.0	1.05	0.0	**0.970**	2.875	1

Table 3-14 LSS Matching Parameters for Various Data Sets to Yield False Alarm Rate ≤ 3

#	Data set	Sensor	Acquisition date	Folder	Window -stride	Hough-bin size	Score-threshold	Prob. of detection	False alarm rate	Rank
1	DRDC-V trial	LW-IR	NA	DRDC_ IR_Data	2	4	3.0	0.481	2.583	13
2	UK trial	LW-IR	04Dec2013	d1	2	4	1.5	0.858	**1.541**	7
				d2	2	4	1.5	0.898	2.416	5
			05Dec2013	d1	2	4	2.0	0.495	2.692	12
				d2	2	4	2.0	0.442	2.957	14
3		SW-IR	04Dec2013	s1	2	4	3.0	0.970	2.597	2
				s2	2	4	2.5	0.953	2.784	3
				s3	2	4	2.5	**1.0**	1.821	1
				s4	2	4	3.0	0.921	2.730	4
				s5	2	4	3.0	0.762	2.607	10
				s6	4	4	2.0	0.859	2.656	6
			05Dec2013	s1	2	4	3.0	0.792	2.401	8
				s2	4	4	4.0	0.758	1.941	11
4	DRDC-V trial	Visible	NA	DRDC_ Vis_Data	4	8	2.0	0.777	2.667	9

THIS PAGE INTENTIONALLY LEFT BLANK

4 CONCLUSIONS

4.1 Concluding Remarks

This report describes the implementation of two feature detectors that are suitable for human detection, namely the HOG SVM and LSS matching algorithms. Moreover, it presents their experimental performance evaluation results using LW-IR, SW-IR as well as visible imagery data. For each algorithm, a set of key detection parameters were varied and the sensitivity of the algorithm's detection results to these variations is illustrated in the ROC curves. In view of the experimental results, we conclude the followings:

- In general, the performance of the HOG SVM and LSS algorithms for human detection using the various DRDC-V and UK trial data sets is promising, indicating that they are suitable for FSAR application. As shown in Table 3-13 and Table 3-14, the detection results are very good in some cases.
- The performance of the HOG SVM algorithm is sensitive not only to the studied key detection parameters but also to the training of the SVM classifier. The selection of a sufficiently comprehensive and heterogeneous set of negative exemplars with a set of ground truth positive exemplars are critical for effectively training the SVM classifier.
- The performance of the LSS matching algorithm is sensitive not only to the studied key detection parameters but also to the number and types of selected query images. More query images often yield higher probability of detection at the cost of higher false alarm rate and computational complexity. In practice, it is expected that a variety of query images capturing different sizes, shapes and poses are used.
- The HOG SVM algorithm runs at near real-time, around 5 to 10 Hz, while LSS matching runs slower.
- It appears that LSS matching yields better detection results than HOG SVM, except for the visible data. However, if a data set contains many different human poses, the number of query images needs to be increased to cover the various poses.
- The results showed that human subjects of sizes as low as 10 × 20 pixels can be detected in some cases. However, a more comprehensive analysis is needed in order to identify the target size limitation of the algorithms more accurately.

4.2 Future Works

All the image sequences processed so far are collected by more or less stationary camera. In order to better assess the performance of the HOG SVM and LSS matching algorithms, we propose the followings to continue this work:

- Obtain new imagery at representative FSAR distances of different backgrounds while the camera is moving to simulate a more realistic ATC data collection system
- Assess the performance of HOG SVM and LSS algorithms using the new imagery data
- For HOG SVM, investigate whether the wide variety in the new imagery would provide a more comprehensive set of negative exemplars along with the true positive exemplars to train the SVM classifier more effectively
- For LSS matching, investigate the optimal approach to create the set of query images

As for any other algorithms, the results of the HOG SVM and LSS human detection algorithms are rarely perfect. Thus, a visual or automated post-processing analysis of the generated detection results, or the use of additional sensor can help validate the detection results. For example, the following automated approaches could eliminate some of the false alarms:

- **Use of temporal information:** Currently, each image frame is processed individually, and hence any false alarms detected on each frame are reported. If the objects are tracked over the frames, only the ones that are detected consistently would be reported.
- **Estimate the dimension of each detection and use a size threshold to remove detections which are too tall, wide or large:** As observed in Figure 3-36, the bounding-boxes of the false alarms stemming from the pole cover only part of the pole. Starting from the detected bounding box, one can apply an automated region-growing approach to estimate the real size of the detected target. This will expand the bounding box vertically to cover the full pole, which would be too tall to be a person.
- **Use additional sensors such as LIDAR to eliminate detections that do not resemble human size:** Detections that appear too tall, too high or too big to be human can be eliminated based on the height and size information provided by LIDAR. This should eliminate some false alarms due to vertical poles, large bushes, top of trees, etc. However, one should be careful not to eliminate true human detections on roof tops, by extracting building and other man-made structures from the scene first.

5 REFERENCES

R-1 S. Se, ATC Task 1 Report – Literature Survey on ATC, RX-RP-53-5690, October 30, 2013.

R-2 M. Ghazel, ATC Task 2 Report – Literature Survey on Facial Recognition, RX-RP-53-5691, November 19, 2013.

R-3 N. Dalal and B. Triggs, Histograms of Oriented Gradients for Human Detection, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2005.

R-4 E. Shechtman and M. Irani, Matching Local Self-Similarities across Images and Videos, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007.

R-5 M. Enzweiler and D.M. Gavrila, Monocular Pedestrian Detection: Survey and Experiments, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 31(12), December 2009.

R-6 OpenCV, http://opencv.org/

R-7 Mexopencv, http://www.cs.stonybrook.edu/~kyamagu/mexopencv/

R-8 LIBSVM, http://www.csie.ntu.edu.tw/~cjlin/libsvm/

R-9 K. Chatfield, J. Philbin and A. Zisserman, Efficient Retrieval of Deformable Shape Classes using Local Self-Similarities, IEEE Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment (NORDIA), Kyoto, Japan, 2009.

THIS PAGE INTENTIONALLY LEFT BLANK

A USER'S GUIDE

A1 Installation & Setup

A1.1 Software Requirements

The following software tools are required:

- Matlab 32-bit or 64-bit version
- Visual Studio (if you need to build the MEX files or compile the C++ code)
- Third-party packages included on delivery DVD. See below for the installation and setup instructions.
    - OpenCV
    - Mexopencv
    - LIBSVM (required for HOG SVM training only)

The following setup has been tested on Matlab R2012a (32-bit and 64-bit) and Matlab R2011a (64-bit), with Visual Studio 2010 on a Windows 7 computer. The pre-built 32-bit MEX files are for Matlab R2012a while the pre-built 64-bit MEX files are for Matlab R2011a. The MEX files may need to be re-built for a different version of Matlab.

A1.2 Third-Party Packages

OpenCV [R-6]

1. Copy OpenCV folder from the delivery DVD to a folder called C:\OpenCV. This folder contains both the 32-bit and 64-bit pre-built binary of OpenCV version 2.4.6.
2. Set up either the 32-bit or 64-bit version that matches your Matlab version:
   - [For 32-bit Matlab] Add `C:\OpenCV\build\x86\vc10\bin` folder to the system PATH environment variable.
   - [For 64-bit Matlab] Add `C:\OpenCV\build\x64\vc10\bin` folder to the system PATH environment variable.

Mexopencv [R-7]

1. Copy mexopencv folder from the delivery DVD to any folder of choice. This folder contains pre-compiled MEX files of mexopencv for both 32-bit and 64-bit Matlab. It includes the SelfSimDescriptor wrapper developed during this project.
2. In Matlab, run the following command to add path:
   - `addpath('C:\...\mexopencv');`
3. Follow the instructions here to re-generate the MEX files if the provided MEX files are not compatible with your Matlab version:
   - Run `mex -setup` in Matlab to select Visual Studio as the compiler if it has not been done before
   - Go to `C:\...\mexopencv` folder in Matlab and run `mexopencv.make`

LIBSVM [R-8]

1. Copy libsvm-3.17 folder from the delivery DVD to any folder of choice. This folder contains pre-compiled MEX files of libsvm for both 32-bit and 64-bit Matlab.
2. In Matlab, run the following command to add path:
   - `addpath('C:\...\libsvm-3.17\matlab');`
3. Follow the instructions here to re-generate the MEX files if the provided MEX files are not compatible with your Matlab version:
   - Run `mex -setup` in Matlab to select Visual Studio as the compiler if it has not been done before
   - Go to `C:\...\libsvm-3.17\matlab` folder in Matlab and run `make`

A1.3 MDA Software

Copy MDA folder from the delivery DVD to any folder of choice. This folder contains the Matlab code as well as C++ code developed by MDA during this project.

- MDA\Matlab\HOGSVM contains documented Matlab code for HOG SVM detector
- MDA\Matlab\LSS contains documented Matlab code for LSS matching
- MDA\Matlab\performance_Evaluation contains Matlab scripts related to performance evaluation
- MDA\HOGSVMdetector contains documented C++ code for HOG SVM detector, together with the Visual Studio 2010 solution/project files and pre-compiled 32-bit and 64-bit executables. You should use the version that matches your OpenCV setup above.
- MDA\LSSmatcher contains documented C++ code for LSS matching, together with Visual Studio 2010 solution/project files and pre-compiled 32-bit and 64-bit executables. You should use the version that matches your OpenCV setup above.

A1.4 Test Data

DRDC-V LW-IR data set in Test_Data\DRDC_IR_Data folder

- Contains 168 infra-red images with ground truth.
- Renamed files by padding zeroes in the filename so that the operating system would sort the images properly (in the raw sub-folder).
- Re-arranged the provided ground truth (imtruth1.mat) to match the new order. The re-arranged ground truth is saved as groundtruth.mat.
- Applied contrast enhancement to the 16-bit raw images to convert into 8-bit PGM files, which are used for testing (in the pgm sub-folder).
- The ground truth info was based on the person head, but HOG SVM training requires the full body. Therefore, an offset is subtracted from the centre of the person head, and a fixed size is used to crop the positive examples. Therefore, this may not encapsulate the entire body if the person gets closer to the camera.
- As the targets are fairly large in this data set, a 48x96 window size is used for HOG.

UK Trial data set in Test_Data\UK_Data folder

- 4_dec subfolder contains 2 LW-IR data sets (d1 and d2) and 6 SW-IR data sets (s1, s2, s3, s4, s5 and s6)
- 5_dec subfolder contains 2 LW-IR data sets (d1 and d2) and 2 SW-IR data sets (s1 and s2)
- As the targets are smaller in these data sets, the LWIR data sets use 32x64 window size for HOG while the SW-IR data sets use 24x48 window size for HOG.

DRDC-V Visible data set in Test_Data\DRDC_Vis_Data folder

- Contains 168 visible images with ground truth
- Renamed files by padding zeroes in the filename so that the operating system would sort the images properly to match against the ground truth
- Fixed the ground truth for image_008.bmp & image_186.bmp, and saved the updated ground truth as info_vis1_fix.mat
- As the targets are large in this data set, a 64x128 window size is used for HOG.

A2 HOG SVM Detector

HOG SVM detector consists of an offline phase of SVM training, followed by run-time detection. It is essential to have many positive and negative representative examples of human for training, which affects the detection performance. The 16-bit raw images together with the ground truth info are used for training, while the 8-bit PGM images are used for testing.

A2.1 Training

HOG SVM training is done in Matlab. The steps are as follows:

```
load groundtruth;

  % groundtruth.mat contains the re-arranged ground truth in a variable
  named info

imagechip = extractpositivechip (foldername, info, interval, offsetx,
offsety, chipwidth, chipheight);

  e.g. imagechip = extractpositivechip (‘C:\Test\raw’, info, [1 1 168],
  20, 20, 40, 80);

  % extract 40x80 positive image chips from each of frame 1 to 168 in
  C:\Test\raw folder using ground truth info, offsetx & offsety are the
  offset of the top left corner of chip relative to ground truth centre

outputimagechip(imagechip, foldername, prefix, width, height);

  e.g. outputimagechip(imagechip, ‘C:\Test\positive’, ‘image’, 48, 96);

  % resize to 48x96 resolution and output image chips to files in
  C:\Test\positive folder with filename prefix ‘image’.  You need to
  create the folder first if it does not exist yet.

imagechip = extractnegativechip(foldername, info, interval,
numperframe, offsetx, offsety, chipwidth, chipheight);

  e.g. imagechip = extractnegativechip(‘C:\Test\raw’, info, [1 1 168],
  2, 20, 20, 40, 80);
  % extract 40x80 resolution negative image chips from each of frame 1
  to 168 in C:\Test\raw folder using ground truth info. Randomly select
  2 negative image chips that do not overlap with any positive image
  chips for each frame

outputimagechip(imagechip, foldername, prefix, width, height);

  e.g. outputimagechip(imagechip, ‘C:\Test\negative’, ‘image’, 48, 96);

  % resize to 48x96 resolution and output image chips to files in
  C:\Test\negative folder with filename prefix ‘image’. You need to
  create the folder first if it does not exist yet.

flip_all(pathPos, pathNeg)

  e.g. flip_all(‘C:\Test\positive’, ‘C:\Test\negative’);

  % flip all the images in C:\Test\positive sideways and save the
  flipped images to the same folder with prefix flipped_, do the same
  for the negative images. This is to increase the number of positive
  and negative examples.

[fpos, fneg]=extractHOGfeatures(pathPos, pathNeg, winsize);

  e.g. [fpos, fneg]=extractHOGfeatures(‘C:\Test\positive’,
  ‘C:\Test\negative’, [48 96]);

  % extract HOG features from all the positive and negative image chips
  using 48x96 as the HOG window size.  This should match the resolution
  of the positive and negative training images.

[model, params] = trainSVM(fpos, fneg, trainparams);

e.g. [model, params] = trainSVM(fpos, fneg, ‘-t 0 –g 0.3 –c 0.5’);

  % perform SVM training on the positive and negative HOG features,
  using the LIBSVM library and the training parameters provided.  –t to
  set type of kernel function, -g to set gamma in kernel function, -c
  to set trade-off between training error and margin

save(‘SVMparams.txt’, ‘params’, ‘-ascii’);

  % output the SVM parameters to file
```

A2.2 Detection

HOG SVM detection can be done in either Matlab or C++.

Matlab:

```
[bboxes, img] = HOGSVMdetector(imagefile, hitthres, scale, finalthres,
winsize, SVMparamfile)

e.g. [bboxes, img] = HOGSVMdetector(‘image_1.pgm’, 0.4, 1.05, 0, [48
96], ‘SVMparams.txt’)
```

- `bboxes: rectangles of the form {[x, y, width, height], …} where humans are detected`
- `img: image with overlaid rectangles corresponding to detection results`
- `imagefile: filename of the test image (8-bit)`
- `hitthres: threshold for distance between features and SVM classifying plane`
- `scale: step size of scales to search`
- `finalthres: final threshold value for detection`
- `winsize: Window size to use for HOG.  This should match the window size used to generate the SVM parameters file during training.`
- `SVMparamfile: SVM param file obtained from SVM training, e.g. SVMparams.txt.  If it is an empty string, it would use the default OpenCV parameters based on the Daimler data set (only if HOG window size is 48x96).`

A test harness Matlab script is also provided to facilitate testing over the whole data set using a particular set of parameters.

```
HOGSVMtestharness(foldername, prefix, ext, startframe, endframe,
hitthres, scale, finalthres, winsize, SVMparamfile, outputvideo)
```

```
e.g. HOGSVMtestharness('C:\Test\pgm', 'image_', 'pgm', 1, 100, 0.4,
1.05, 0, [48 96], 'SVMparams.txt', 'results.avi')
```

which would repeatedly invoke HOGSVMdetector above and generate an output video file. If outputvideo is an empty string, no video will be created.

C++:

The source code, Visual Studio 2010 project/solution as well as executable files are provided. The 32-bit executable is located in the Release sub-folder while the 64-bit executable is located in the x64\Release sub-folder. Use the executable on the command line as follows:

```
HOGSVMdetector.exe imagefile output.txt output.jpg hitthres scale
finalthres winwidth winheight SVMparamfile
```

```
e.g. HOGSVMdetector.exe image_1.pgm output.txt output.jpg 0.4 1.05 0
48 96 SVMparams.txt
```

- `imagefile: filename of the test image (8-bit)`
- `output.txt: output text file with the rectangles corresponding to the detection results.  Each line contains x, y, width, height for each detection.`
- `output.jpg: output image file with the overlaid rectangles corresponding to the detection results`
- `hitthres: threshold for distance between features and SVM classifying plane`

- `scale: step size of scales to search`
- `finalthres: final threshold value for detection`
- `winwidth: width for HOG window`
- `winheight: height for HOG window`
- `SVMparamfile: SVM param file obtained from SVM training, e.g. SVMparams.txt.  If omitted, it would use the default OpenCV parameters based on the Daimler data set (only if HOG window size is 48x96).`

A3 LSS Matching

LSS matching can be done in either Matlab or C++.

Matlab:

`[bboxes, img] = LSSmatcher(queryimagefile, testimagefile, winstride, houghbinsize, scorethres)`

```
e.g. [bboxes, img] = LSSmatcher('query.pgm', 'image_1.pgm', 4, 4, 2.0)
```

- `bboxes: rectangles of the form {[x, y, width, height], …} where the query image are detected`
- `img: image with overlaid rectangles corresponding to the detection results`
- `queryimagefile: filename of the query image (8-bit)`
- `testimagefile: filename of the test image (8-bit)`
- `winstride: stride for how dense the uniform sampling is`
- `houghbinsize: Hough transform bin size`
- `scorethres: score threshold value for detection`

A test harness Matlab script is also provided to facilitate testing over the whole data set using a particular set of parameters.

`LSStestharness(queryimagefile, foldername, prefix, extm startframe, endframe, winstride, houghbinsize, scorethres, outputvideo)`

```
e.g. LSStestharness('query.pgm', 'C:\Test\pgm', 'image_', 'pgm', 1,
100, 4, 4, 2.0, 'results.avi')
```

which would repeatedly invoke LSSmatcher above and generate an output video file. If outputvideo is an empty string, no video will be created.

If there are multiple query images, instead of invoking one by one for each query image, another Matlab script for batch mode can be used, where the LSS descriptors for the test image are only computed once and hence more computationally efficient. Moreover, bounding boxes that overlap substantially with each other are filtered out.

[bboxes, img] = LSSmatcher_batch(queryimagefileArray, testimagefile, winstride, houghbinsize, scorethres)

```
e.g. [bboxes, img] = LSSmatcher_batch({‘query.pgm’,‘query2.pgm’},
‘image_1.pgm’, 4, 4, 2.0)
```

- bboxes: rectangles of the form {[x, y, width, height], …} where any of the query images is detected
- img: image with overlaid rectangles corresponding to the detection results for all the query images
- queryimagefileArray: cell array of filenames of the query images (8-bit)

Similarly, a test harness Matlab script is also provided to facilitate testing over the whole data set using a particular set of parameters with multiple query images.

LSStestharness_batch(queryimagefileArray, foldername, prefix, ext, startframe, endframe, winstride, houghbinsize, scorethres, outputvideo)

```
e.g. LSStestharness_batch({‘query.pgm’,’query2.pgm’}, ‘C:\Test\pgm’,
‘image_’, ‘pgm’, 1, 100, 4, 4, 2.0, ‘results.avi’)
```

C++:

The source code, Visual Studio 2010 project/solution as well as executable files are provided for single query image LSS matching. The 32-bit executable is located in the Release sub-folder while the 64-bit executable is located in the x64\Release sub-folder. Run the executable on the command line as follows:

LSSmatcher.exe queryimagefile testimagefile output.txt output.jpg winstride hougbinsize scorethres

```
e.g. LSSmatcher.exe query.pgm image_1.pgm output.txt output.jpg 4 4
2.0
```

- queryimagefile: filename of the query image (8-bit)
- testimagefile: filename of the test image (8-bit)
- output.txt: output text file with the rectangles corresponding to the detection results. Each line contains x, y, width, height for each detection.
- output.jpg: output image file with the overlaid rectangles corresponding to the detection results
- winstride: stride for how dense the uniform sampling is
- houghbinsize: Hough transform bin size
- scorethres: score threshold value for detection

A4 Performance Evaluation

Matlab scripts are provided to facilitate performance evaluation and generate ROC curves automatically, for both HOG SVM and LSS matching.

A4.1 HOG SVM

Matlab execution command:

- `[ status ] = HOG_SVM_ROC_Analysis( HOG_SVM_ROC_config_file )`

Input:

- `HOG_SVM_ROC_config_file`: A self-explanatory input configuration text file, which contains all the input configurations, settings and parameters needed to conduct the HOG SVM ROC analysis. A template of this file is provided with the Matlab code.

Return value:

- `status`: An execution status flag (1: success and 0: failure)

A4.2 LSS Matching

Matlab execution command:

- `[ status ] = LSS_ROC_Analysis( LSS_ROC_config_file )`

Input:

- `LSS_ROC_config_file`: A self-explanatory input configuration text file, which contains all the input configurations, settings and parameters needed to conduct the LSS ROC analysis. A template of this file is provided with the Matlab code. Note that the configuration files corresponding to the HOG SVM and LSS algorithms are different and hence specific to their respective algorithm.

Return value:

- `status`: An execution status flag (1: success and 0: failure)

A4.3 Saved Output

For both the HOG SVM and the LSS algorithms, the following files and folders may be saved in the output directory, as specified in the input configuration file:

- `config_params_struct.mat`: A Matlab file containing the structure of the all configuration input parameters read from the input configuration file.

- `results_struct.mat`: A Matlab file containing the structure of HOG SVM key detection parameters values iterations at each image and processing image and the corresponding detection results for each combination of detection parameters.
- `roc_results_array.mat`: A Matlab file containing an array, where the first column is the false alarm rate and the second column is the probability of detection for each iteration.
- `roc_best_results.mat`: A Matlab file containing an array, detailing the best detection results and their corresponding detection parameters values, which meet a maximum false alarm rate, as specified in the configuration file.
- **`figures`**: This is a folder containing JPEG files of the generated figures, such as those presented in this report, if the user selects to save these figures in the configuration file.
- **`detection_images`**: This is a folder containing JPEG files of each image with the overlaid detections. Use this only if all the algorithm detection parameters are fixed. These detection images are saved only if the user selects this option in the configuration file.
- **`negative_examples`**: This is a folder containing PGM image files of all the detected false alarms. Use this only if all the algorithm detection parameters are fixed. These false alarm examples images are saved only if the user selects this option in the configuration file.