A Project Report on

Small Target Detection in Infrared Images

Submitted for partial fulfilment of award of

Bachelor of Technology

In

Computer Science & Engineering

By

Piyush Bhatt (1202113)

Shubham Sisodia (1202133)

Shreya Dariyal (1202131)

Under the supervision of

Dr. Shashi Kant Verma

Head of Department (Assistant Professor),

Computer Science and Engineering Department

GOVIND BALLABH PANT ENGINEERING COLLEGE, PAURI

June, 2016

CANDIDATE'S DECLARATION

We hereby declare that the work which is being presented in this report entitled, **"Small Target Detection in Infrared Images",** submitted towards the partial fulfilment of the requirement of the degree of **Bachelor of Technology (Computer Science and Engineering)** is an authentic record of our own work carried out under the supervision and guidance of Dr. S. K. Verma, Department of Computer Science and Engineering, G. B. Pant Engineering College, Pauri-Garhwal, Uttarakhand.

I have not submitted the matter embodied in this thesis report for the award of any other degree.

Piyush Bhatt (1202113)
Shubham Sisodia (1202133)
Shreya Dariyal (1202131)

Department of Computer Science and Engineering
G. B. Pant Engineering College
Ghurdauri, Pauri Garhwal-246194 (Uttarakhand)
(An Autonomous College Of Government of Uttarakhand)

CERTIFICATE

This is to certify that this dissertation entitled **"Small Target Detection in Infrared Images"** submitted by **Piyush Bhatt (1202113), Shubham Sisodia (1202133) and Shreya Dariyal (1202131)** to the Department of Computer Science and Engineering, in partial fulfillment of requirements for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering**, during Final year **(Aug 2015-June 2016)** is an authentic record of the work carried out by them under our supervision and guidance.

We wish them success for their future endeavors.

Dr. S. K. Verma	Mr. Ramesh Kumar	Dr. S. K. Verma
(Project Guide)	(Project Coordinator)	(Head, CSED)

ABSTRACT

The robust detection of small targets is one of the key techniques in infrared search and tracking applications. A novel small target detection method in a single infrared image is implemented in this project. Initially, the traditional infrared image model is generalized to a new infrared patch-image model using local patch construction. Then, because of the non-local self-correlation property of the infrared background image, based on the new model small target detection is formulated as an optimization problem of recovering low-rank and sparse matrices, which is effectively solved using stable principle component pursuit. Finally, a simple adaptive segmentation method is used to segment the target image and the segmentation result can be refined by post-processing. Extensive synthetic and real data experiments show that under different clutter backgrounds the proposed method not only works more stably for the varying number of targets, but also has the better detection performance compared with conventional baseline methods.

ACKNOWLEDGEMENT

Ab initio, we express our gratitude and appreciation to our venerated project guide Dr. S. K. Verma who have been utmost benign, affable and reasonable toward us and as the H.O.D. generated a genial atmosphere in the department, conducive for carrying out our project work in a blithesome manner.

Special thanks and hearted sense of gratitude to Mr. R. Kumar, Dr. H. S. Bhaduria, Dr. Bhumika Gupta, Mr. Y. Bhandari, Mr. Ashwini Saini and Mr. Abhishek Gupta for their sonorous support. Without their help our project would be a figment.

Last but not the least we are grateful and indebted to our parents, teachers and seniors and all those persons who have in some way or the other helped us in various stages of our lives.

Piyush Bhatt (1202113)
Shubham Sisodia (1202133)
Shreya Dariyal (1202131)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 MOTIVATION

An infra-red search and track (IRST) system (sometimes known as infra-red sighting and tracking) is a method for detecting and tracking objects which give off infrared radiation such as jet aircraft and helicopters [15]. IRST is a generalized case of forward looking infrared (FLIR), i.e. from forward-looking to all-round situation awareness. Such systems are passive (thermographic camera), meaning they do not give out any radiation of their own, unlike radar. This gives them the advantage that they are difficult to detect [16].

1.2 OBJECTIVE

Infrared small target detection is one of the key techniques in infrared search and track (IRST) systems. IRST (Infrared Search and Track) has been applied to many military or civil fields such as precise guidance, aerospace, forest early warning, remote sensing and so on, but the detection becomes difficult task because infrared targets have their own characteristics, such as target size variation. The performance of the whole IRST system depends on the accuracy of detection results. When there exists heavy noise and clutter such as cloud clutter and sea clutter, small targets are usually buried in a complex background with low signal-to-clutter ratio (SCR). Moreover, small targets have no concrete shape and texture because of the long imaging distance. Therefore, small target detection in complex infrared background is considered a difficult and challenging problem. Although many research efforts have been focused on this area in past decades, it still remains an open problem.

1.3 INTRODUCTION

1.3.1 INFRARED

Infrared (IR) is invisible radiant energy, electromagnetic radiation with longer wavelengths than those of visible light, extending from the nominal red edge of the visible spectrum at 700 nanometres (frequency 430 THz) to 1 mm (300 GHz) (although people can see infrared up to at least 1050 nm in experiments). Most of the thermal radiation emitted by objects near room temperature is infrared.

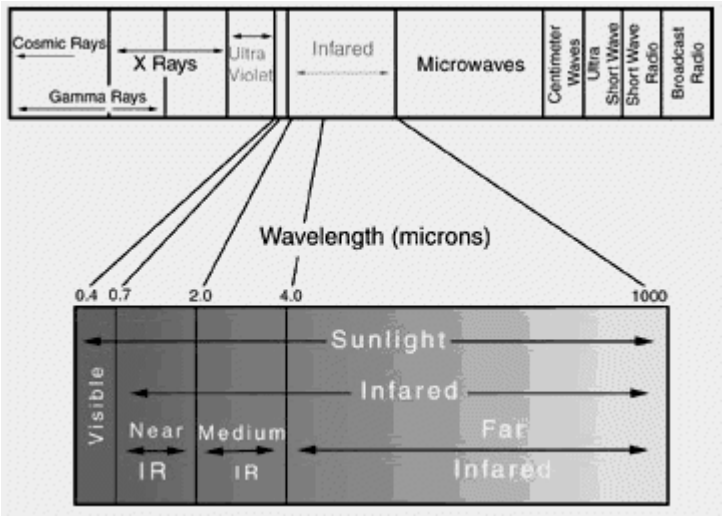


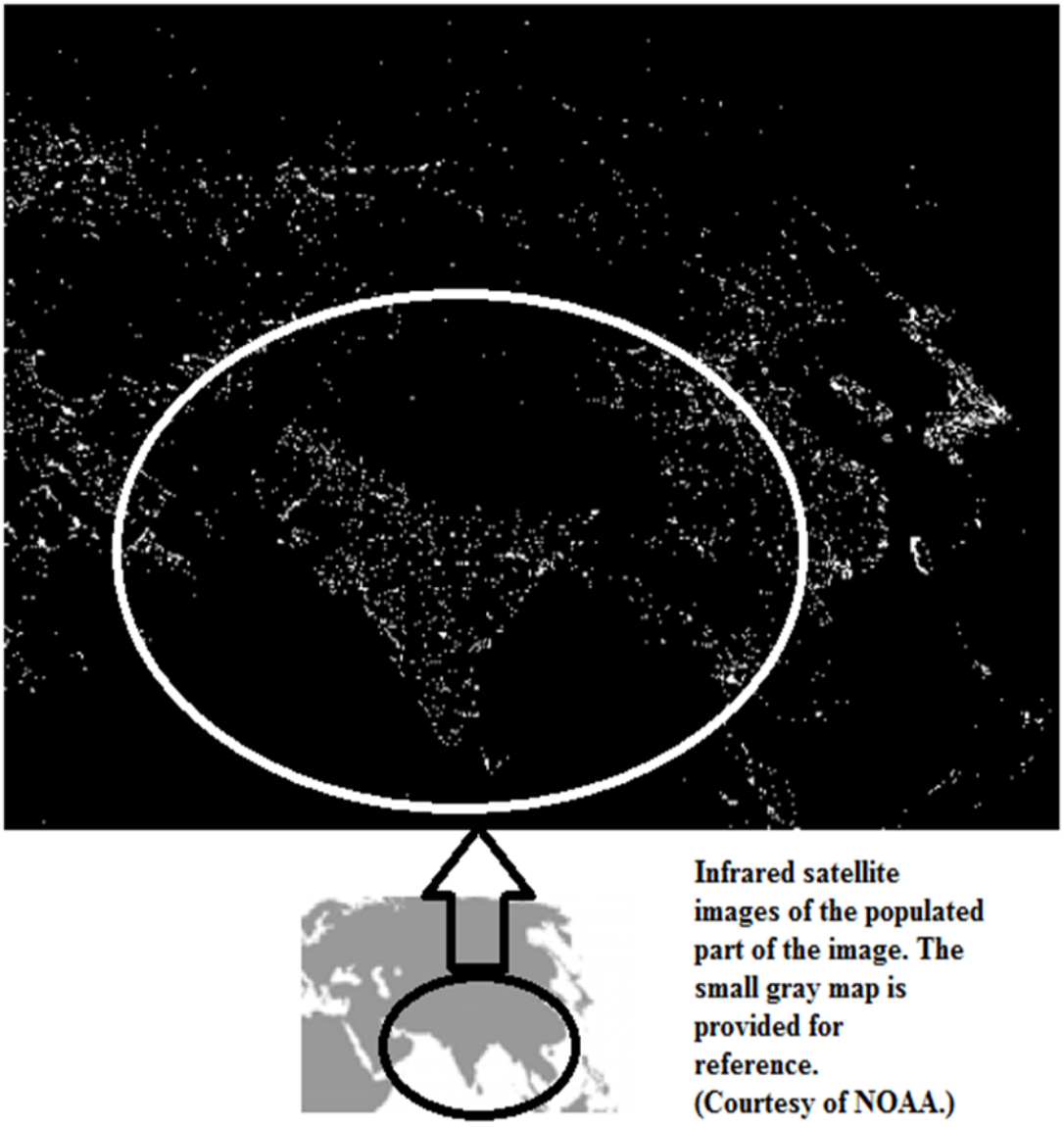
Figure 1.1 Spectrum showing infrared range [10]

Infrared radiation was discovered in 1800 by astronomer Sir William Herschel, who discovered a type of invisible radiation in the spectrum lower in energy than red light, by means of its effect upon a thermometer. Slightly more than half of the total energy from the Sun was eventually found to arrive on Earth in the form of infrared. The balance between absorbed and emitted infrared radiation has a critical effect on Earth's climate.

Infrared energy is emitted or absorbed by molecules when they change their rotational-vibrational movements. Infrared energy excites vibrational modes in a molecule through a change in the dipole moment, making it a useful frequency range for study of these energy states for molecules of the proper symmetry. Infrared spectroscopy examines absorption and transmission of photons in the infrared energy range.

Band No.	Name	Wavelength (μm)	Characteristics and Uses
1	Near infrared	0.76–0.90	Biomass and shoreline mapping
2	Middle infrared	1.55–1.75	Moisture content of soil and vegetation
3	Thermal infrared	10.4–12.5	Soil moisture; thermal mapping
4	Middle infrared	2.08–2.35	Mineral mapping

Regions within infrared with characteristics and uses



Infrared satellite images of the populated part of the image. The small gray map is provided for reference. (Courtesy of NOAA.)

Figure 1.2. Infrared Characteristics (above) and Infrared Satellite Image (below).

NATURAL INFRARED

Sunlight, at an effective temperature of 5,780 kelvins, is composed of nearly thermal-spectrum radiation that is slightly more than half infrared. At zenith, sunlight provides an irradiance of just over 1 kilowatt per square meter at sea level. Of this energy, 527 watts is infrared radiation, 445 watts is visible light, and 32 watts is ultraviolet radiation.

On the surface of Earth, at far lower temperatures than the surface of the Sun, almost all thermal radiation consists of infrared in various wavelengths. Of these natural thermal radiation processes only lightning and natural fires are hot enough to produce much visible energy, and fires produce far more infrared than visible-light energy.

1.3.2 INFRA-RED SEARCH AND TRACK

The first use of an IRST system appears to be the F-101 Voodoo, F-102 Delta Dagger and F-106 Delta Dart interceptors. The F-106 had an early IRST mounting replaced in 1963 with a production retractable mount. The IRST was also incorporated into the Vought F-8 Crusader (F-8E variant) which allowed passive tracking of heat emissions and was similar to the later Texas Instruments AAA-4 installed on early F-4 Phantoms.

These were fairly simple systems consisting of an infra-red sensor with a horizontally rotating shutter in front of it. The shutter was slaved to a display under the main interception radar display in the cockpit, any IR light falling on the sensor would generate a "pip" on the display, in a fashion similar to the B-scopes used on early radars. The display was primarily intended to allow the radar operator to manually turn the radar to the approximate angle of the target, in an era when radar systems had to be "locked on" by hand. The system was considered to be of limited utility, and with the introduction of more automated radars they disappeared from fighter designs for some time. IRST systems re-appeared on more modern designs starting in the 1980s with the introduction of 2-D sensors, which cued both horizontal and vertical angle.

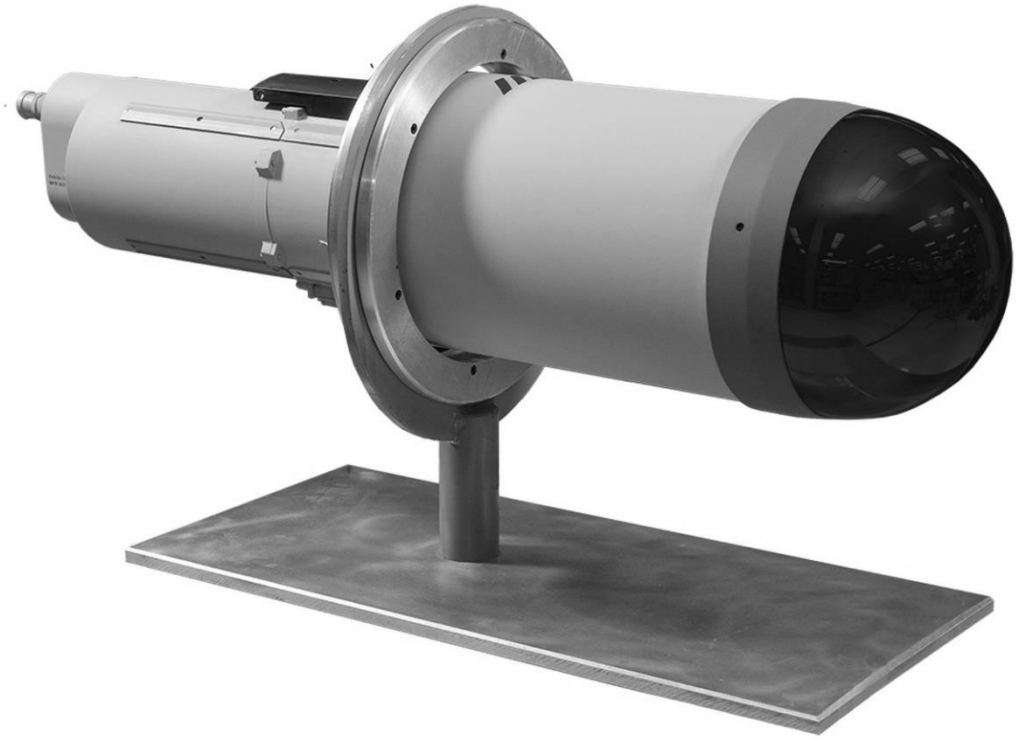


Figure 1.3. IRST system real life model

Sensitivities were also greatly improved, leading to better resolution and range. In more recent years, new systems have entered the market. In 2015, Northrop Grumman introduced its OpenPod(TM) IRST pod, which uses a sensor by Selex ES.

Figure 1.4. Hatsu officer served Aircraft with IRST [11]

1.3.3 SEQUENTIAL AND SINGLE FRAME DETECTION METHODS

The sequential detection methods are processed on the basis of the prior information of the target and background. Thus, these methods cannot achieve satisfactory performance because information can hardly be obtained in military applications. Moreover, the capability of sequential methods commonly depends on the results of single frame detection.

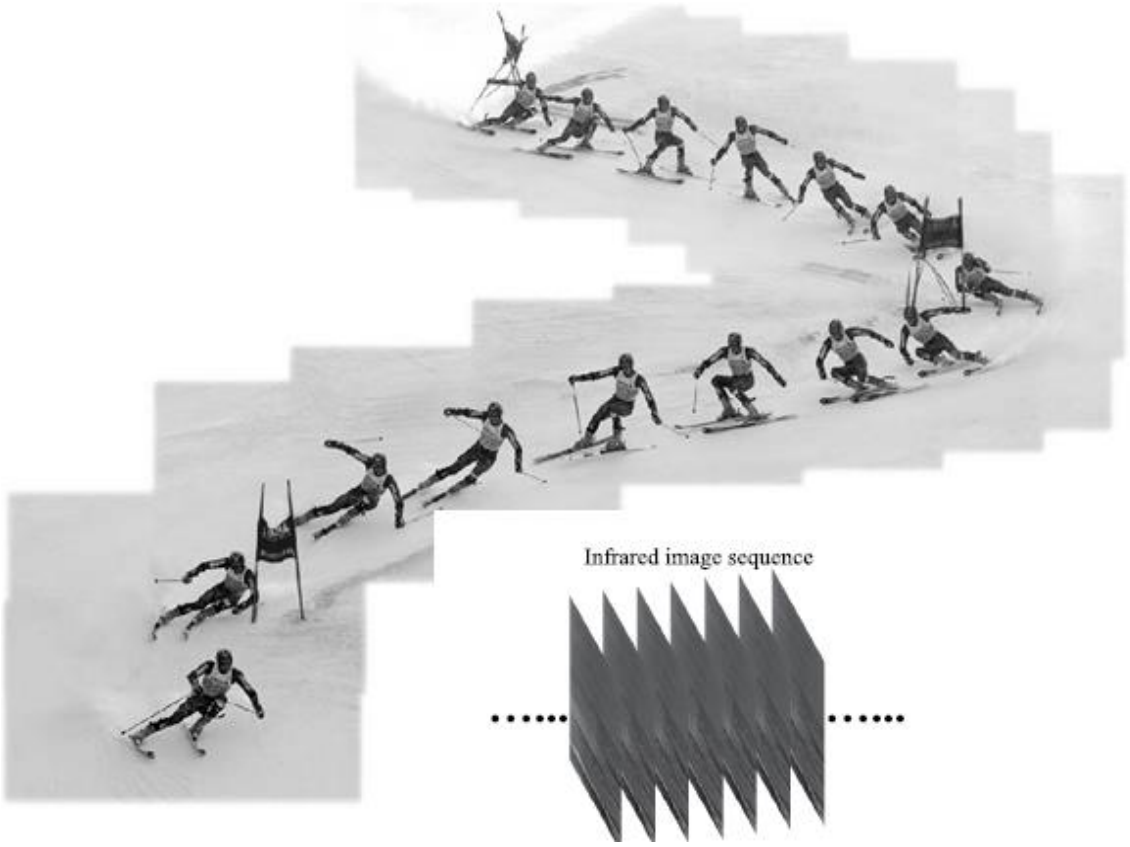


Figure 1.5. Sequential frames

In practical applications where the IRST systems need to search and track the targets, the imaging backgrounds usually change fast since many targets, such as anti-ship missiles, high speed airplanes and so on, have very high velocities.

CHAPTER 2
LITERATURE REVIEW

1) Optical moving target detection with 3D matched filtering

S. Reed, R. M. Gagliardi, and L. B. Stotts introduced one of the early classical method, 3D matched (directional) filtering [1] which is so called track-before-detection (TBD) technique. This method can detect moving constant-velocity targets, where the knowledge of the target shape and velocity is required. The concept is based on electronic digital signal processing following imaging of the target scene on a CCD array. The array produces a sequence of data frames and the 3D algorithms produce filtered images in which the targets emerge as observable tracks. Devices needed to successfully implement the algorithms are discussed, and basic shortfalls and anomalies are considered. Key elements appear to be the spatial light modulators and stored read-outs of holograms or storage plates.

2) Analysis of multiframe target detection using pixel statistics

P. Wei, B. Zeidler, and W. Ku presented a method in which the 3D directional filter method can be considered as a path-based statistic, a pixel-based statistic for the analysis of multi-frame target detection [2]. This allows the TBD approach to be characterized as an image enhancement algorithm with detection gains compared with single frame detections. It is shown that for the TBD approach to have superior detection over single frame detection the target signal-to-noise ratio (SNR) must be greater than a threshold SNR in order to overcome the uncertainty in the target path. Tradeoffs are made for a class of velocity constrained target paths in terms of the detection gain with respect to the maximum target velocity and number of frames integrated.

3) Tracking point targets in cloud clutter

J. Silverman, J. M. Mooney, and C. E. Caefer proposed a temporal profile method for this small target detection problem considering this problem in a 1D space. The generic temporal filter is a damped sinusoid, implemented recursively. Their final algorithm, a triple temporal filter (TTF) [3] based on six parameters, consists of a

sequence of two damped sinusoids followed by an exponential averaging filter, with an edge suppression feature. Three 'trackers' based on the TTF operate in real-time hardware on laboratory IR cameras including: an empirical initial version; and two recent forms identified by an optimization routine. The latter two operate best in the two distinct realms: one for evolving cloud clutter, the other for temporal nose-dominated scenes such as blue sky or stagnant clouds. Results are presented both as specific examples and metric plots over an extensive database of local scenes with targets of opportunity.

4) *Fast new small-target detection algorithm based on a modified partial differential equation in infrared clutter*

Biyin Zhang, Tianxu Zhang, Zhiguo Cao and Kun Zhang proposed an adaptive anisotropic filter based on a modified partial differential equation (AFMPDE) [6] to detect a small target in such a strong cluttered background. A regularizing operator is employed to adaptively eliminate structured background and simultaneously enhance the target signal. The proposed algorithm's performance is illustrated and compared with the two-dimensional least mean square (TDLMS) adaptive filter on real IR image data. Experimental results demonstrate that the proposed novel method is fast and effective.

5) *Temporal profile based small moving target detection algorithm in infrared image sequences*

D. Liu, J. Zhang, and W. Dong, proposed a new algorithm employing the connecting line of the stagnation points (CLSP) of the temporal profile as the baseline [4]. The deviation of the temporal profile and its CLSP is analyzed and it is determined that the distribution of the residual temporal profile obtained by subtracting the baseline from the temporal profile can be modeled by a Gaussian distribution. The occurrences of the targets have intensity values significantly different to the distribution of the residual temporal profile. Unlike the conventional 3-D method, this new algorithm operates on the temporal profile in 1-D space, not in 3-D space, thus having a higher computational efficiency. Experiments with real IR image sequences have proved the validity of the new approach.

6) Small target detection using bilateral filter and temporal cross product in infrared images

Tae-Wuk Bae introduced a spatial and temporal target detection method using spatial bilateral filter (BF) and temporal cross product (TCP) of temporal pixels in infrared (IR) image sequences [5]. At first, the TCP is presented to extract the characteristics of temporal pixels by using temporal profile in respective spatial coordinates of pixels. The TCP represents the cross product values by the gray level distance vector of a current temporal pixel and the adjacent temporal pixel, as well as the horizontal distance vector of the current temporal pixel and a temporal pixel corresponding to potential target center. The summation of TCP values of temporal pixels in spatial coordinates makes the temporal target image (TTI), which represents the temporal target information of temporal pixels in spatial coordinates. And then the proposed BF filter is used to extract the spatial target information. In order to predict background without targets, the proposed BF filter uses standard deviations obtained by an exponential mapping of the TCP value corresponding to the coordinate of a pixel processed spatially. The spatial target image (STI) is made by subtracting the predicted image from the original image. Thus, the spatial and temporal target image (STTI) is achieved by multiplying the STI and the TTI, and then targets finally are detected in STTI.

7) Infrared small target detection based on modified local entropy and EMD

H. Deng, J. G. Liu, and Z. Chen presented a method for Infrared small target detection based on modified local entropy and EMD [10]**.** Image entropy and empirical mode decomposition (EMD) are effective methods for target detection. EMD algorithm is a powerful tool for adaptive multiscale analysis of nonstationary signals. A new technique based on EMD and modified local entropy is proposed in small target detection under sea-sky background. With the EMD algorithm, it is valid to estimate the background and get the target image by removing the background from the original image and segmenting the target based on the modified local entropy method.

8) Performance evaluation of 2D adaptive prediction filters for detection of small objects in image data

T. Soni, J. R. Zeidler, and W. H. Ku studied the performance of two dimensional least mean square (TDLMS) [7] adaptive filters as prewhitening filters for the detection of small objects in image data. The object of interest is assumed to have a very small spatial spread and is obscured by correlated clutter of much larger spatial extent. The correlated clutter is predicted and subtracted from the input signal, leaving components of the spatially small signal in the residual output. The receiver operating characteristics of a detection system augmented by a TDLMS prewhitening filter are plotted using Monte-Carlo techniques. It is shown that such a detector has better operating characteristics than a conventional matched filter in the presence of correlated clutter. For very low signal to background ratios, TDLMS based detection systems show a considerable reduction in the number of false alarms.

9) Enhanced detectability of small objects in correlated clutter using an improved 2D adaptive lattice algorithm

P. A. Ffrench, J. R. Zeidler, and W. H. Ku proposed an improved 2-D adaptive lattice algorithm [8] and its application to the removal of correlated clutter to enhance the detectability of small objects in images. The two improvements proposed here are increased flexibility in the calculation of the reflection coefficients and a 2-D method to update the correlations used in the 2-D AL algorithm. The 2-D AL algorithm is shown to predict correlated clutter in image data and the resulting filter is compared with an ideal Wiener-Hopf filter. The results of the clutter removal will be compared to previously published ones for a 2-D least mean square (LMS) algorithm. 2-D AL is better able to predict spatially varying clutter than the 2-D LMS algorithm, since it converges faster to new image properties.

10) Small target detection using two dimensional least mean square (TDLMS) filter based on neighbourhood analysis

Y. Cao, R. Liu, and J. Yang proposed a Small target detection using two dimensional least mean square (TDLMS) filter based on neighbourhood analysis [9]. TDLMS is a general adaptive filter algorithm and when applied to infrared small target detection, traditional structure and implementation of TDLMS may cause some

problems in this field. This paper presented a new TDLMS filter structure and implementation incorporating neighborhood analysis and data fusion, which is capable of acquiring and analyzing more information from the vicinity of the target, leading to a more prominent detection result. This enables TDLMS filter to perform better and become more suitable in the field of small target detection.

11) Morphology-based algorithm for point target detection in infrared backgrounds

V. Tom, T. Peli, M. Leung, and J. Bondaryk proposed a Morphology-based algorithm [11] for point target detection in infrared backgrounds. It exhibits comparable detection and false alarm performance to a median filter. The morphology-based algorithm has an efficient computational paradigm based on combinations of simple nonlinear grayscale operations, which makes it ideally suited to real- time, high data rate IRST applications. A detection filter based on morphological background estimation exhibits spatial high-pass characteristics emphasizing target-like peaks in the data and suppressing all other clutter. Example cases are presented which point out the detection performance differences between the morphological and median approaches.

12) Modified top-hat transformation based on contour structuring element to detect infrared small target

X. Bai, F. Zhou, Y. Xie, and T. Jin presented a modified top-hat transformation [12] based on contour structuring element to detect infrared small target. Top-hat transformation has been widely used for infrared small target detection, but it is sensitive to the clutter and may destroy the details of the image, which make it an ineffective way for small target detection. To detect the infrared small target simply and efficiently, a new Top-hat transformation based on contour structuring element is proposed and then modified according to the property of the target region to enhance and detect the infrared small target in this paper.

13) Max-mean and Max-median filters for detection of small-targets

S. Deshpande, M. Er, V. Ronda, and P. Chan proposed Max-mean and Max-median filters for detection of small-targets [13]. This deals with the problem of detection and tracking of low observable small-targets from a sequence of IR images

against structural background and non-stationary clutter. In this paper, they investigated the usefulness of Max-Mean and Max-Median filters in preserving the edges of clouds and structural backgrounds, which helps in detecting small-targets. Subsequently, anti-mean and anti-median operations result in good performance of detecting targets against moving clutter. The raw image is first filtered by max-mean/max-median filter. Then the filtered output is subtracted from the original image to enhance the potential targets. A thresholding step is incorporated in order to limit the number of potential target pixels. The threshold is obtained by using the statistics of the image. Finally, the thresholded images are accumulated so that the moving target forms a continuous trajectory and can be detected by using the post-processing algorithm. It is assumed that most of the targets occupy a couple of pixels. Head-on moving and maneuvering targets are not considered.

14) Optimal spatial adaptation for patch based image de-noising

C. Kervrann and J. Boulanger have used a novel adaptive and patch-based approach for image denoising and representation [39]. The method is based on a pointwise selection of small image patches of fixed size in the variable neighborhood of each pixel. Their contribution is to associate with each pixel the weighted sum of data points within an adaptive neighborhood, in a manner that it balances the accuracy of approximation and the stochastic error, at each spatial position. This method is general and can be applied under the assumption that there exists repetitive patterns in a local neighborhood of a point. By introducing spatial adaptivity, they have extend the work earlier described by Buades et al. which can be considered as an extension of bilateral filtering to image patches. Finally, they propose a nearly parameter-free algorithm for image denoising. The method is applied to both artificially corrupted (white Gaussian noise) and real images and the performance is very close to, and in some cases even surpasses, that of the already published denoising methods

15) A frequency domain algorithm for multi-frame detection and estimation of dim targets

B. Porat and B. Friedlander have proposed an algorithm for detecting moving targets by imaging sensors and estimating their trajectories is proposed [16]. The algorithm is based on directional filtering in the frequency domain, using a bank of filters for all possible target directions. The directional filtering effectively integrates

the target signal, resulting in an improved signal-to-noise ratio. Working in the frequency domain facilitates a considerable reduction in computational requirements compared to time-domain algorithms. The algorithm is described in detail, and its false alarm and detection probabilities are analysed.

16) *Adaptive mean and variance filter for detection of dim point-like targets*

E. T. Lim, L. Shue, and R. Venkateswarlu have presented this paper for the problem of detecting small target in IR imagery has attracted much research effort over the past few decades [22]. As opposed to early detection algorithms which detect targets spatially in each image and then apply tracking algorithm, more recent approaches have used multiple frames to incorporate temporal as well as spatial information. They often referred to as track before detect algorithms. This approach has shown promising results particularly for detection of dim point-like targets. However, the computationally complexity has prohibited practical usage for such algorithms. This paper presents an adaptive, recursive and computation efficient detection method. This detection algorithm updates parameters and detects occurrence of targets as new frame arrived without storing previous frames, thus achieved recursively. Besides, the target temporal intensity change is modelled by two Gaussian distribution with different mean and variance. The derivation of this generalized model has taken account of the wide variation of target speed, therefore detects wider range of targets.

17) *Novel dim target detection and estimation algorithm based on double threshold partial differential equation*

M. Li, T. X. Zhang, Z. R. Zuo, X. C. Sun, and W. D. Yang have presented the present work to propose a brand-new algorithm based on an adaptive double threshold nonlinear anisotropic diffusion equation (DTPDE) to detect and track moving dim targets against complex cluttered background in infrared (IR) image sequences. They also illustrate the performance comparisons of the proposed algorithm DTPDE and two-dimensional least mean squares (TDLMS) on real IR image sequence data. Extensive experiment results demonstrate the proposed novel algorithm's flexibility and adaptability in detecting moving weak dim targets

18) Infrared Patch-Image Model for Small Target Detection in a Single Image

Chenqiang Gao, Deyu Meng, Yi Yang, Yongtao Wang, Xiaofang Zhou and Alexander G. Hauptmann proposed an Infrared Patch-Image Model for Small Target Detection [14] in a Single Image. A novel small target detection method in a single infrared image is proposed in this paper. Initially, the traditional infrared image model is generalized to a new infrared patch-image model using local patch construction. Then, because of the non-local self-correlation property of the infrared background image, based on the new model small target detection is formulated as an optimization problem of recovering low-rank and sparse matrices, which is effectively solved using stable principle component pursuit. Finally, a simple adaptive segmentation method is used to segment the target image and the segmentation result can be refined by post-processing. Extensive synthetic and real data experiments show that under different clutter backgrounds the proposed method not only works more stably for different target sizes and signal-to-clutter ratio values, but also has better detection performance compared with conventional baseline methods.

CHAPTER 3
PROBLEM DEFINITION

3.1 INFRARED IMAGE MODEL

Generally, the infrared image model can be formulated as follows:

$$f_D(x, y) = f_T(x, y) + f_B(x, y) + f_N(x, y),$$

where f_D, f_T, f_B, f_N, and (x, y) are the original infrared image, the target image, the background image, the random noise image and the pixel location, respectively. Depending on whether the focus is on the target image, the background image or both of them leads to different methods.

Some conventional small target detection methods, such as TopHat filtering [10], [11] and so on, focused on background image f_B and used filters to predict the background image in order to suppress the background clutter. These methods can usually achieve good performance in applications. However, they are sensitive to noise and usually could not work stably when the target size varies within a somewhat large range [4].

Generally speaking, small target detection methods are usually based on some assumptions on the target, the background or both of them. The suitability of the assumption would determine the robustness of the corresponding method in applications.

Detection range varies with:

- clouds
- altitude
- air temperature
- target's attitude
- target's speed

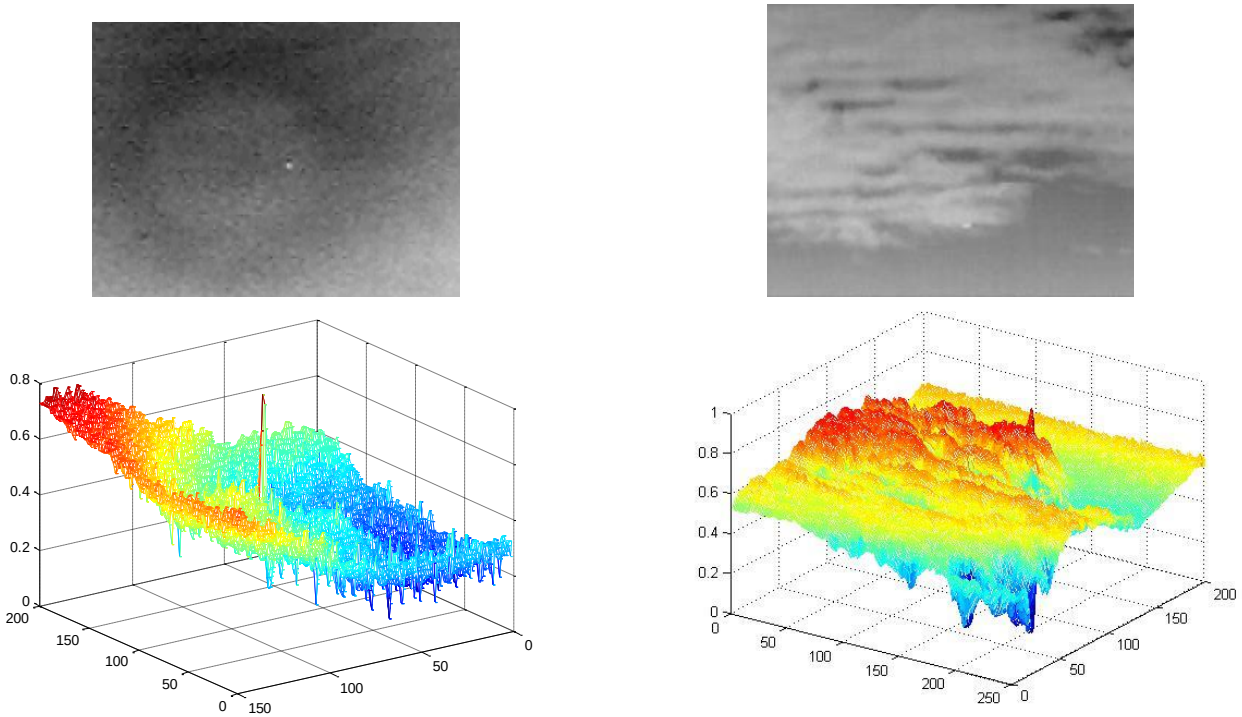


Figure 3.1. Representative real Infrared Images containing a small target (upper) and the corresponding 3D surfaces (lower) in different backgrounds.

3.2 INFRARED PATCH-IMAGE MODEL FOR SMALL TARGET DETECTION

3.2.1 PATCH IMAGE MODEL

Given the original image, the background image, the target image and the noise image, we can construct the corresponding patch-images D, B, T and N, respectively. Then the traditional infrared image model is converted to a new one called Infrared Patch-image (IPI) model: **D = B + T + N**

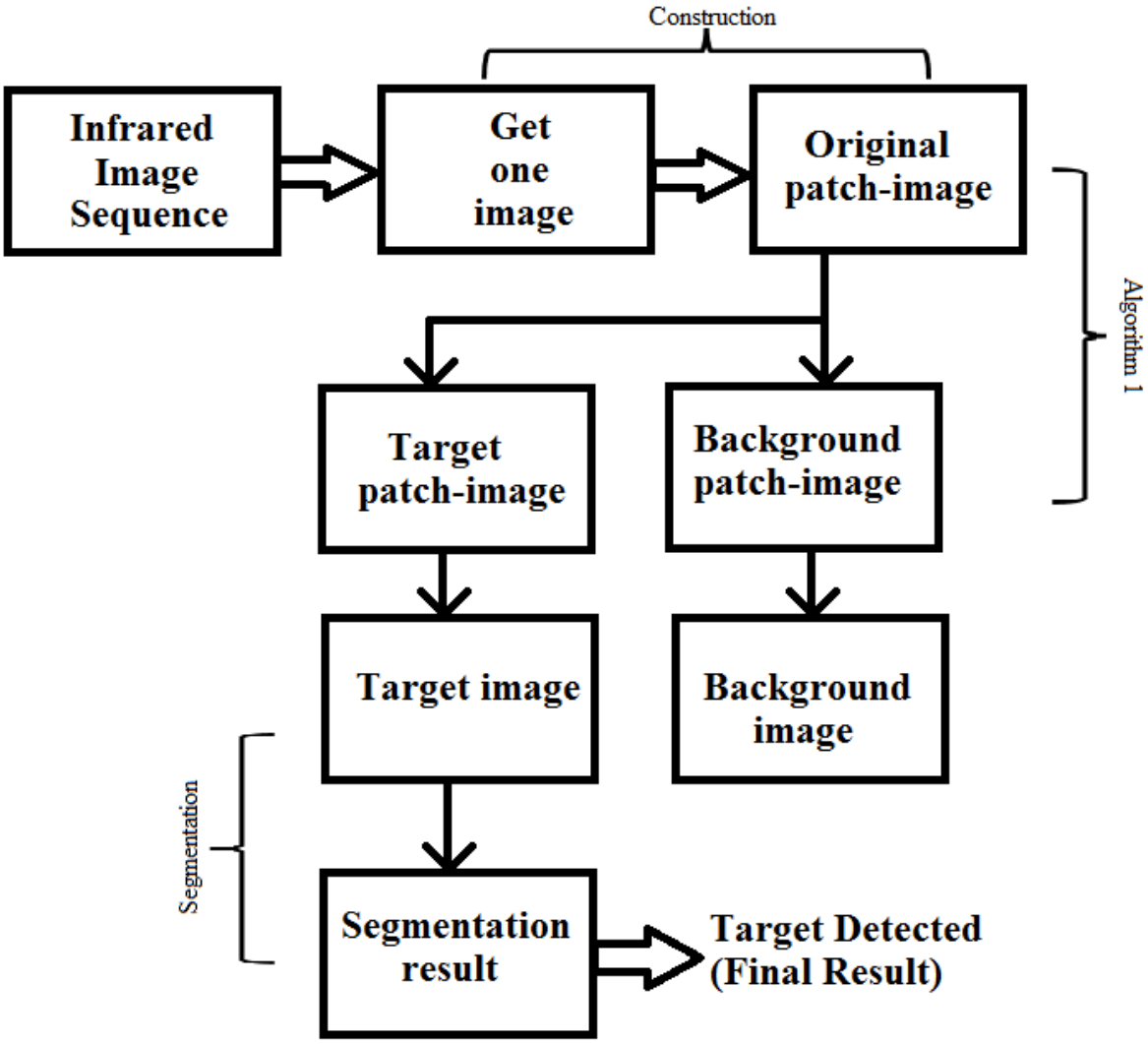


Figure 3.2. Block diagram

In the above shown block diagram, Algorithm 1 represents the optimal algorithm which is to be chosen for solving the optimization problem which has been proven to be convex in nature in the reference [14]. Reference [14] uses Accelerated Proximal Gradient (APG) approach proposed in [17].

3.2.2 PROPERTIES OF PATCH IMAGES

A. Target Patch-Image T

In practical applications, a small target usually keeps changing all the time. The brightness may vary from dim to bright and its size may vary from 4 × 4 to more than 10 × 10 (in pixel). However, it is small with respect to the whole image, thus the target image can be considered as a sparse matrix, which makes the corresponding target patch-image T still be a sparse matrix. That is:

$$\| \boldsymbol{T} \|_0 < \boldsymbol{k}$$

where $\| \cdot \|_0$ denotes the $\boldsymbol{l_0}$ norm which counts the number of nonzero entries, and k is determined by the number of small targets and their sizes (support), and apparently $\boldsymbol{k} << \boldsymbol{m} \times \boldsymbol{n}$ ($m \times n$ is the size of T), which means that most of the entries of the matrix T are zero. Besides this sparseness property, we do not make any additional assumption on the target image.

B. Background Patch-Image B

Due to atmospheric refraction, dispersion, optical defocusing, lens aberration, diffraction, deformation of mirror, and detector tilt, the original infrared background image tends to be slightly blurred [17], [18] and many local patches are approximately linearly correlated with each other even though the pixel distance between two patches may be large in an image. This property of non-local self-correlation exists commonly in infrared background images.

Thus this background patch image B can be considered as a low rank matrix

$$\textbf{Rank (B)} \leq \textbf{r,}$$

where r is a constant. Intrinsically, r constrains the complexity of the background image and the value of r is larger for the complex background than for the uniform background.

In our context, the background is generally with a non-local self-correlation configuration. In such cases, one low-rank subspace cluster is generally enough to model the background situations. We thus only employ one low-rank subspace assumption in our model to make it with a more concise form.

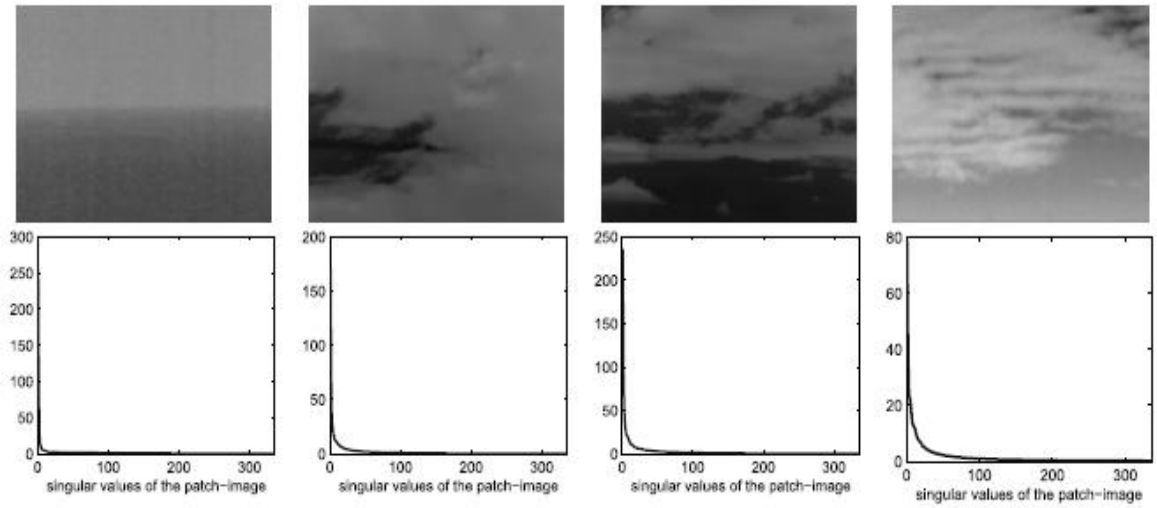


Figure 3.3 The low-rank property of the background patch-images. The first row are four representative background images and the second one are the singular values of the corresponding background patch-images [14].

C. Noise Patch-Image N

$\|N\|_F \leq \delta$ for some $\delta > 0$. Thus we have:

$$\|D \mid\!- B - T\|_F \leq \delta,$$

where $\|\cdot\|_F$ is the Frobenius norm (i.e. $\|X\|_F = \sqrt{\sum_{ij} X_{ij}^2}$). [14]

Although the parameters k, r and δ in (5), (6) and (7) vary for different infrared images, it is not necessary to compute their values directly. [14]

3.2.3 CONSTRUCTION AND RECONSTRUCTION

Figure 4 shows the construction of a patch-image from an image. Firstly, a series of local image patches can be obtained by using a sliding window from left and top to right and down in an image. Then we give vector representation of each patch as a column of a new image (matrix). In this way, we can convert an image to a new image which is called a *patch-image.* For some special cases, such as when the size of the sliding window is the same as the original image or the sliding window equals the column of the original image and the horizontal sliding step is 1, the patch-image is the same as the original image. How to choose the sliding steps and the sliding window, namely the patch size, depends on the specific applications.

Figure 5 shows the reconstruction of an image from a patch image. Because the local patches usually overlap each other, a pixel location in reconstructed image would correspond to several values from different patches. Thus, we should define a 1D filter function to determine the pixel value *v*.

$$v = f(\mathbf{x}),$$

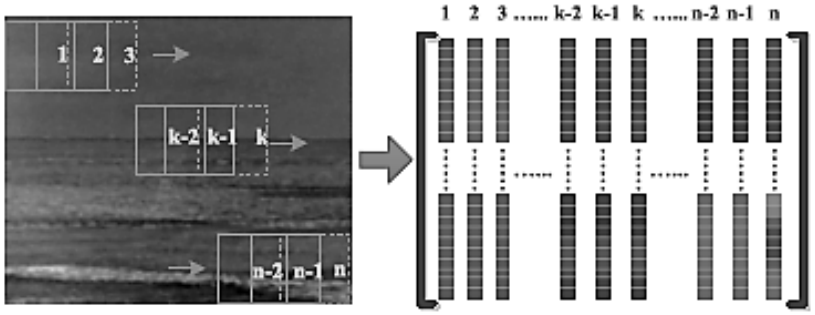


Figure 3.4 Constructing the patch-image from an original image by using image patches. (Left) An original image. (Right) The patch- image.

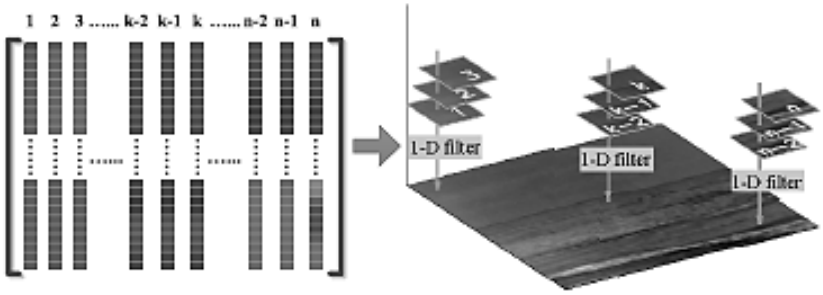


Figure 3.5 Reconstructing the image from a patch-image. (Left) The patch-image. (Right) The reconstructed image.

The definition of the 1D filter function f depends on the practical applications and we can define it as:

v = median(x),

v = mean(x),

v = max(x), or

v = min(x) and so on.

3.3 TARGET IMAGE SEGMENTATION

3.3.1 CONVEX OPTIMIZATION PROBLEM

In computer vision, image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as super-pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyse, which in this case is the separation of the target from image background. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images, which in this case is the target. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics.

First using the construction step mentioned in 1.4.2 the patch image D is constructed from the original image f_{D} which is a single image taken out of the image sequence. Then we apply Algorithm 1 to this patch image D to estimate simultaneously the low-rank background patch-image B and the sparse target-image T. Third, we reconstruct the background image and target image from the patch-images B and T, respectively.

3.3.2 POST PROCESSING

In the final step, some post-processing techniques can be used to refine the segmentation results. For example, we can use region analysis techniques to delete false detections or use morphological techniques to refine the target regions. In addition, for any detected target we can use statistical techniques to estimate the complexity of its corresponding local region in the reconstructed background image, and then the estimated result can be used to evaluate its reliability. For example, if the corresponding local background of a target is simple, we could believe that the detection target is more reliable and vice versa. Generally speaking, the post-processing is still an important step to achieve good detection results.

CHAPTER 4
PROPOSED METHOD

4.1 PLATFORM USED

MATLAB was used for implementing the new Infrared patch image model, convex optimization and for performing the segmentation on the set of 5 real images and the synthetic images which will be generated in the later part of the project. MATLAB was preferred because of its following advantages over other languages:

- A very large (and growing) database of built-in algorithms for image processing and computer vision applications
- MATLAB allows you to test algorithms immediately without recompilation. You can type something at the command line or execute a section in the editor and immediately see the results, greatly facilitating algorithm development.
- The MATLAB Desktop environment, which allows you to work interactively with your data, helps you to keep track of files and variables, and simplifies common programming/debugging tasks
- The ability to read in a wide variety of both common and domain-specific image formats.
- The ability to call external libraries, such as OpenCV
- Clearly written documentation with many examples, as well as online resources such as web seminars ("webinars").
- Bi-annual updates with new algorithms, features, and performance enhancements
- If you are already using MATLAB for other purposes, such as simulation, optimization, statistics, or data analysis, then there is a very quick learning curve for using it in image processing.
- Technical support from a well-staffed, professional organization (assuming your maintenance is up-to-date)
- A large user community with lots of free code and knowledge sharing
- The ability to auto-generate C code, using MATLAB Coder, for a large (and growing) subset of image processing and mathematical functions, which you could then use in other environments, such as embedded systems or as a component in other software.

4.2 PROPOSED METHOD OVERVIEW

First, the patch-image D is constructed from the original infrared image f_D which is obtained from an image sequence. Second, Algorithm 1 is applied to the patch-image D to estimate simultaneously the low-rank background patch-image B and the sparse target patch-image T. Third, we reconstruct the background image f_B and target image f_T from the patch-images B and T , respectively.

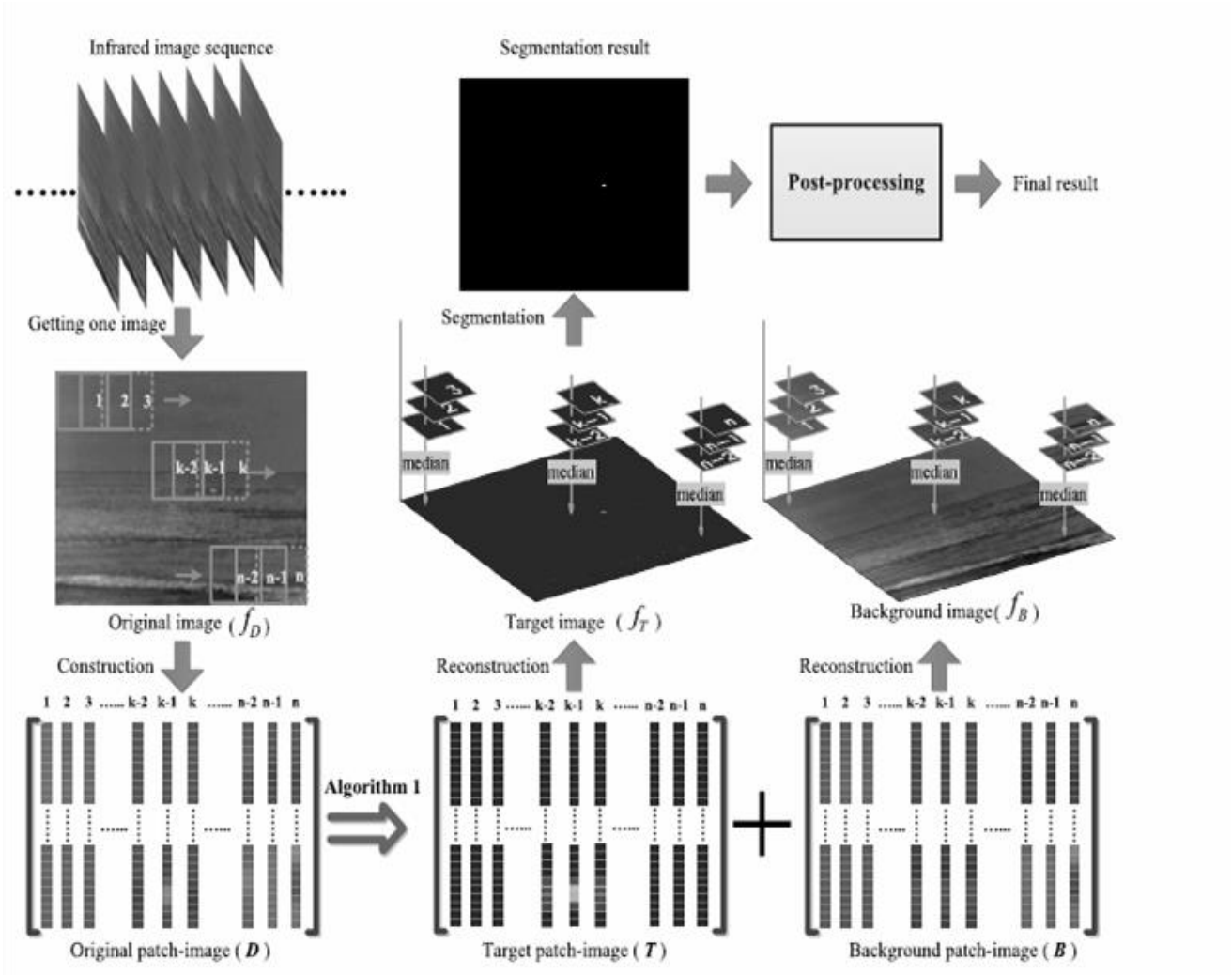


Figure 4.1. The overview of the proposed method

4.3 PATCH IMAGE CONSTRUCTION

4.3.1 PATCH IMAGE PARAMETERS

First step in the implementation of the infrared patch image construction is the definition of the patch window. Defining the patch windows needs the definition of the variables which represents the four following parameters:

```
6 -    param.dw = 50;          %window width
7 -    param.dh = 50;          %window height
8 -    param.x_step = 10;      %horizontal step
9 -    param.y_step = 10;      %vertical step
```

Figure 4.2. Window parameter definition

Window width

Width of the patch window in pixels.

Variable: param.dw

Window height:

Height of the patch window in pixels.

Variable: param.dh

Horizontal sliding step

Horizontal pixel difference between two successive windows.

Variable: param.x_step

Vertical sliding step:

Pixel difference between two windows in successive rows.

Variable: param.y_step

In the above implementation we have set the window size to be 50x50 with the horizontal sliding step of 10 pixels and the vertical sliding step of 10 pixels. All these parameters are stored in a single vector named **'param',** which along with the infrared image under processing is sent as the parameter to the function which constructs the patch image and will do the further processing.

4.3.2 CONSTRUCTION

We call a function named '**winRPCA_median (I, param)' is** called for every image from the main function. '**I'** contains the infrared image and the '**param'** is a vector containing the parameters for the patch window.

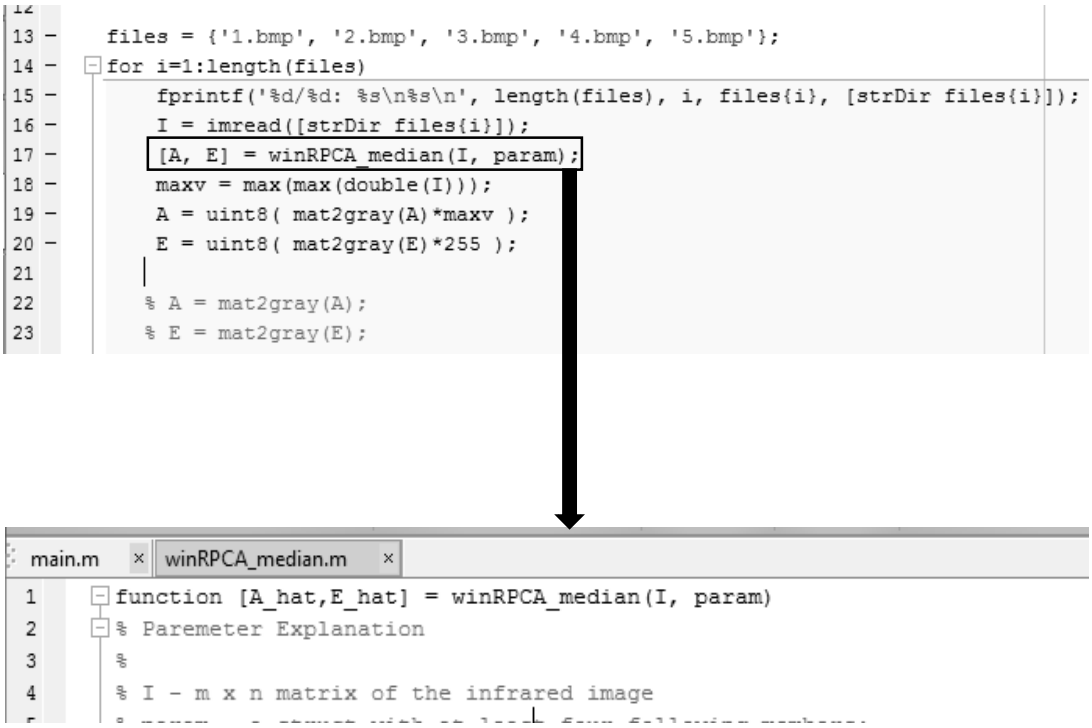


Figure 4.3. Main program Calling 'winRPCA_median' function

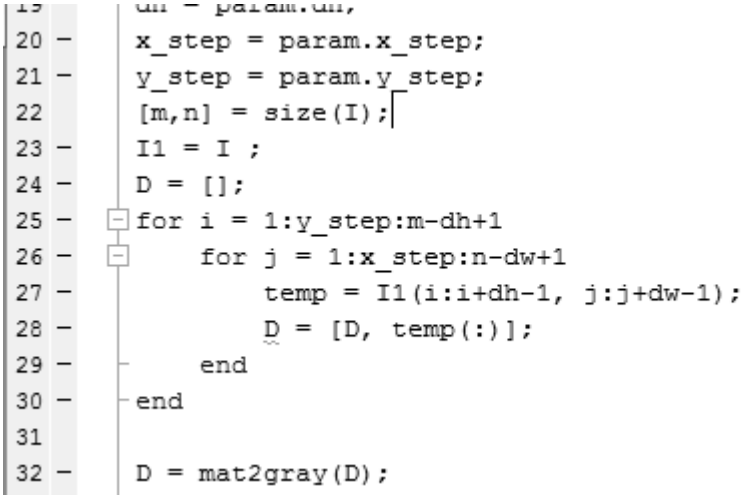
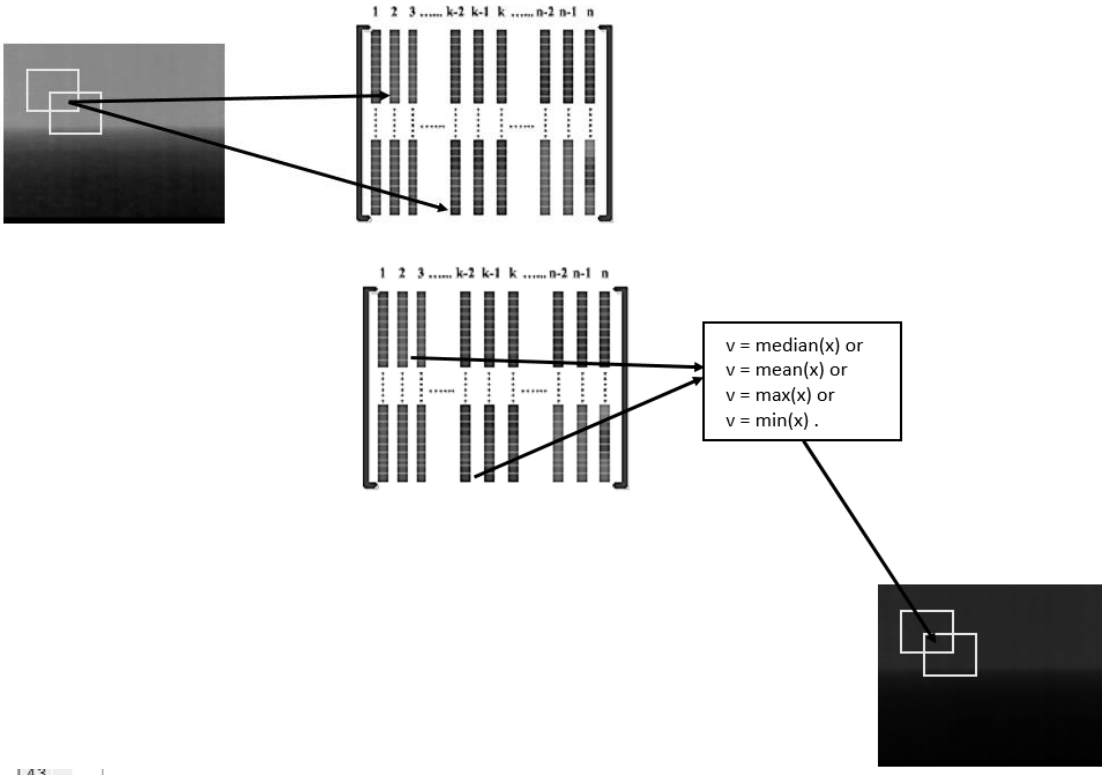


Figure 4.4. Construction of patch image inside winRPCA_median

4.3.3 RECONSTRUCTION

In the reconstruction step we form the image from a given patch image. But the problem is that more than one value might correspond to the same pixle so we are using the median to overcome this confusion.




```
43
44      %reconstruction using median
45 -    for i = 1:y_step:m-dh+1
46 -        for j = 1:x_step:n-dw+1
47 -            index = 1+index;
48 -            temp(:) = A1(:, index);
49 -            C(i:i+dh-1, j:j+dw-1) = C(i:i+dh-1, j:j+dw-1)+1;
50 -            temp1(:) = E1(:, index);
51 -            for ii = i:i+dh-1
52 -                for jj = j:j+dw-1
53 -                    AA(ii,jj, C(ii,jj)) = temp(ii-i+1, jj-j+1);
54 -                    EE(ii,jj, C(ii,jj)) = temp1(ii-i+1, jj-j+1);
55 -                end
56 -            end
57 -        end
58 -    end
59
60 -    for i=1:m
61 -        for j=1:n
62 -            if C(i,j) > 0
63 -                A_hat(i,j) = median(AA(i,j,1:C(i,j)));
64 -                E_hat(i,j) = median(EE(i,j,1:C(i,j)));
65 -            end
66 -        end
67 -    end
68
```

Figure 4.5. (Above) Reconstruction problem and (below) Implementation

4.4 CONVEX OPTIMIZATION ALGORITHMS

In computer vision, image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as super-pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyse, which in this case is the separation of the target from image background. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images, which in this case is the target. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics.

First using the construction step mentioned in 1.4.2 the patch image D is constructed from the original image f_D which is a single image taken out of the image sequence. Then we apply Algorithm 1 to this patch image D to estimate simultaneously the low-rank background patch-image B and the sparse target-image T. Third, we reconstruct the background image and target image from the patch-images B and T, respectively.

The estimation of the low-rank background patch-image B and the sparse target-image T from the patch image D has been proved to be a convex optimization problem [14]. Here Algorithm 1 represents the algorithm for solving this convex problem and is a separate topic of research. Reference [14] has solved this problem using the method of **Accelerated Proximal Gradient (APG)** proposed in [19] which goes as follows:

Algorithm 1 Solution via Accelerated Proximal Gradient

Input: Infrared patch-image matrix $D \in R^{m\times n}$, λ.

$B_0 = B_{-1} = 0; T_0 = T_{-1} = 0; a_0 = a_{-1} = 1; \mu_0 > 0; \bar{\mu} > 0; \eta < 1.$

while not converged **do**

$Y_k^B = B_k + \frac{a_{k-1}-1}{a_k}(B_k - B_{k-1}), Y_k^T = T_k + \frac{a_{k-1}-1}{a_k}(T_k - T_{k-1}).$

$G_k^B = Y_k^B - \frac{1}{2}(Y_k^B + Y_k^T - D).$

$(U, S, V) = \text{svd}(G_k^B), B_{k+1} = U\mathcal{S}_{\frac{\mu_k}{2}}[S]V^t.$

$G_k^T = Y_k^T - \frac{1}{2}(Y_k^B + Y_k^T - D).$

$T_{k+1} = \mathcal{S}_{\frac{\lambda\mu_k}{2}}[G_k^T].$

$a_{k+1} = \frac{1+\sqrt{4a_k^2+1}}{2}; \mu_{k+1} = \max(\eta\mu_k, \bar{\mu}).$

$k = k + 1.$

end while

Output: $B = B_k, T = T_k.$

Augmented Lagrange Multiplier (ALM) Method

The basic form of the exact ALM function is [A, E] = exact_alm_rpca(D, λ), and that of the inexact ALM function is [A, E] = inexact_alm_rpca(D, λ), where D is a real matrix and λ is a positive real number. We solve the RPCA problem using the method of augmented Lagrange multipliers. The method converges Q-linearly to the optimal solution. The exact ALM algorithm is simple to implement, each iteration involves computing a partial SVD of a matrix the size of D, and converges to the true solution in a small number of iterations.

Algorithm 1 (RPCA via the Exact ALM Method)

Input: Observation matrix $D \in \mathbb{R}^{m \times n}$, λ.

$Y_0^* = \text{sgn}(D)/J(\text{sgn}(D))$; $\mu_0 > 0$; $\rho > 1$; $k = 0$.
while not converged do
  // Lines 4-12 solve $(A_{k+1}^*, E_{k+1}^*) = \arg\min_{A,E} L(A, E, Y_k^*, \mu_k)$.
  $A_{k+1}^0 = A_k^*$, $E_{k+1}^0 = E_k^*$, $j = 0$;
  while not converged do
    // Lines 7-8 solve $A_{k+1}^{j+1} = \arg\min_A L(A, E_{k+1}^j, Y_k^*, \mu_k)$.
    $(U, S, V) = \text{svd}(D - E_{k+1}^j + \mu_k^{-1} Y_k^*)$;
    $A_{k+1}^{j+1} = U \mathcal{S}_{\mu_k^{-1}}[S] V^T$;
    // Line 10 solves $E_{k+1}^{j+1} = \arg\min_E L(A_{k+1}^{j+1}, E, Y_k^*, \mu_k)$.
    $E_{k+1}^{j+1} = \mathcal{S}_{\lambda\mu_k^{-1}}[D - A_{k+1}^{j+1} + \mu_k^{-1} Y_k^*]$;
    $j \quad j + 1$.
  end while
  $Y_{k+1}^* = Y_k^* + \mu_k (D - A_{k+1}^* - E_{k+1}^*)$; $\mu_{k+1} = \rho \mu_k$.
  $k \quad k + 1$.
end while

Output: (A_k^*, E_k^*).

The algorithm can be further speeded up by using a fast continuation technique, thereby yielding the modified ALM algorithm known as the inexact ALM algorithm, which goes as follows:

Algorithm 1 (RPCA via the Inexact ALM Method)

Input: Observation matrix $D \in \mathbb{R}^{m \times n}$, λ.

1: $Y_0 = D/J(D)$; $E_0 = 0$; $\mu_0 > 0$; $\rho > 1$; $k = 0$.
2: **while** not converged **do**
3: // Lines 4-5 solve $A_{k+1} = \arg\min_A L(A, E_k, Y_k, \mu_k)$.
4: $(U, S, V) = \text{svd}(D - E_k + \mu_k^{-1} Y_k)$;
5: $A_{k+1} = U \mathcal{S}_{\mu_k^{-1}}[S] V^T$.
6: // Line 7 solves $E_{k+1} = \arg\min_E L(A_{k+1}, E, Y_k, \mu_k)$.
7: $E_{k+1} = \mathcal{S}_{\lambda \mu_k^{-1}}[D - A_{k+1} + \mu_k^{-1} Y_k]$.
8: $Y_{k+1} = Y_k + \mu_k (D - A_{k+1} - E_{k+1})$; $\mu_{k+1} = \rho \mu_k$.
9: $k \quad k + 1$.
10: **end while**

Output: (A_k, E_k).

Dual Method

The most basic form of the function is [A, E] = dual_rpca(D, λ), where D is a real matrix and λ is a positive real number.

Algorithm 1 (Robust PCA via the Dual)

Input: Observation matrix $D \in \mathbb{R}^{m \times n}$, λ.

1: $Y_0 = \text{sgn}(D)/J(\text{sgn}(D))$; $k \leftarrow 0$.
2: **while** not converged **do**
3: Compute the projection D_k of D onto $N(Y_k)$:
4: **if** $\|Y_k\|_2 > \lambda^{-1} |Y_k|_\infty$ **then**
5: $D_k \leftarrow \pi_2(D)$, $A \leftarrow D$, $E \leftarrow 0$.
6: **else if** $\lambda^{-1} |Y_k|_\infty > \|Y_k\|_2$ **then**
7: $D_k \leftarrow \pi_\infty(D)$, $A \leftarrow 0$, $E \leftarrow D$.
8: **else**
9: $A \leftarrow 0$, $E \leftarrow 0$.
10: **while** not converged **do**
11: $A \leftarrow \pi_2(D - E)$, $E \leftarrow \pi_\infty(D - A)$.
12: **end while**
13: $D_k \leftarrow A + E$.
14: **end if**
15: Do line search to determine a step size δ_k.
16: $Y_{k+1} \leftarrow \frac{Y_k + \delta_k (D - D_k)}{J(Y_k + \delta_k (D - D_k))}$ and $k \leftarrow k + 1$.
17: **end while**

Output: (A, E).

We solve the convex dual of the RPCA problem, and retrieve the low-rank and sparse error matrices from the dual optimal solution. The algorithm computes only a partial SVD in each iteration and hence, scales well with the size of the matrix D

Singular Value Thresholding

The most basic form of the function is [A, E] = singular_value_rpca(D, λ), where D is a real matrix and λ is a positive real number. Here again, we solve a relaxation of the original RPCA problem, albeit different from the one solved by the Accelerated Proximal Gradient (APG) method. The algorithm is extremely simple to implement, and the computational complexity of each iteration is about the same as that of the APG method. However, the number of iterations to convergence is typically quite large.

Algorithm 1: Singular Value Thresholding (SVT) Algorithm

Input: sampled set Ω and sampled entries $\mathcal{P}_\Omega(M)$, step size δ, tolerance ϵ, parameter τ, increment ℓ, and maximum iteration count $k_{\max}$
Output: X^{opt}
Description: Recover a low-rank matrix M from a subset of sampled entries

Set $Y^0 = k_0 \delta \mathcal{P}_\Omega(M)$ (k_0 is defined in (5.3))
Set $r_0 = 0$
for $k = 1$ **to** $k_{\max}$
    Set $s_k = r_{k-1} + 1$
    repeat
        Compute $[U^{k-1}, \Sigma^{k-1}, V^{k-1}]_{s_k}$
        Set $s_k = s_k + \ell$
    until $\sigma^{k-1}_{s_k - \ell} \le \tau$
    Set $r_k = \max\{j : \sigma^{k-1}_j > \tau\}$
    Set $X^k = \sum_{j=1}^{r_k} (\sigma^{k-1}_j - \tau) u^{k-1}_j v^{k-1}_j$

    if $\|\mathcal{P}_\Omega(X^k - M)\|_F / \|\mathcal{P}_\Omega M\|_F \le \epsilon$ **then break**

    Set $Y^k_{ij} = \begin{cases} 0 & \text{if } (i,j) \notin \Omega, \\ Y^{k-1}_{ij} + \delta(M_{ij} - X^k_{ij}) & \text{if } (i,j) \in \Omega \end{cases}$
end *for* k
Set $X^{\text{opt}} = X^k$

4.5 CONVEX OPTIMIZATION IMPLEMENTATION

First using the construction step mentioned in 4.2.1 the parameters for patch window are defined, then using the steps mentioned in 4.2.2 patch image D is constructed from the original image I which is a single image taken out of the image sequence. Then we need to apply Algorithm 1 to this patch image D to estimate simultaneously the low-rank background patch-image A and the sparse target-image E. All the algorithms mentioned in the 3.3 are implemented and the optimization result and the time taken by these algorithms for this convex optimization problem is represented in the following section.

4.5.1 SINGULAR VALUE THRESHOLDING

We can solve convex optimization by calling singular_value_rpca (…) from the winRPCA (…).

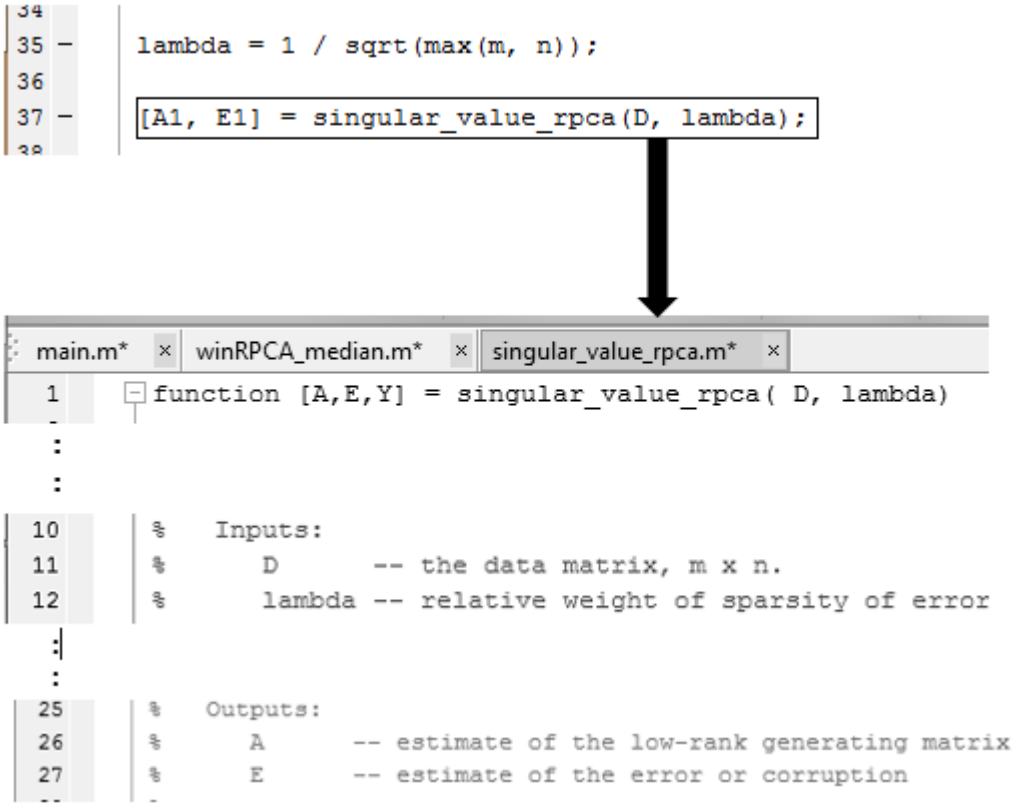


Figure 4.6. Calling SVT from winRPCA_median and parameter explanation.

4.5.2 ACCELERATED PROXIMAL GRADIENT(APG)

We can solve convex optimization by calling APG_IR(…) from the winRPCA (…). In this the iteration process can be finished if the rank of patch image A keeps the same many times. In practice, the APG algorithm, sometimes, cannot get a strictly low-rank matrix after iteration process. Many singular values of the obtained matrix, however, are extremely small. This can be considered to be low-rank to a certain extent. Experimental results show that the final recovered background image and target image are good.

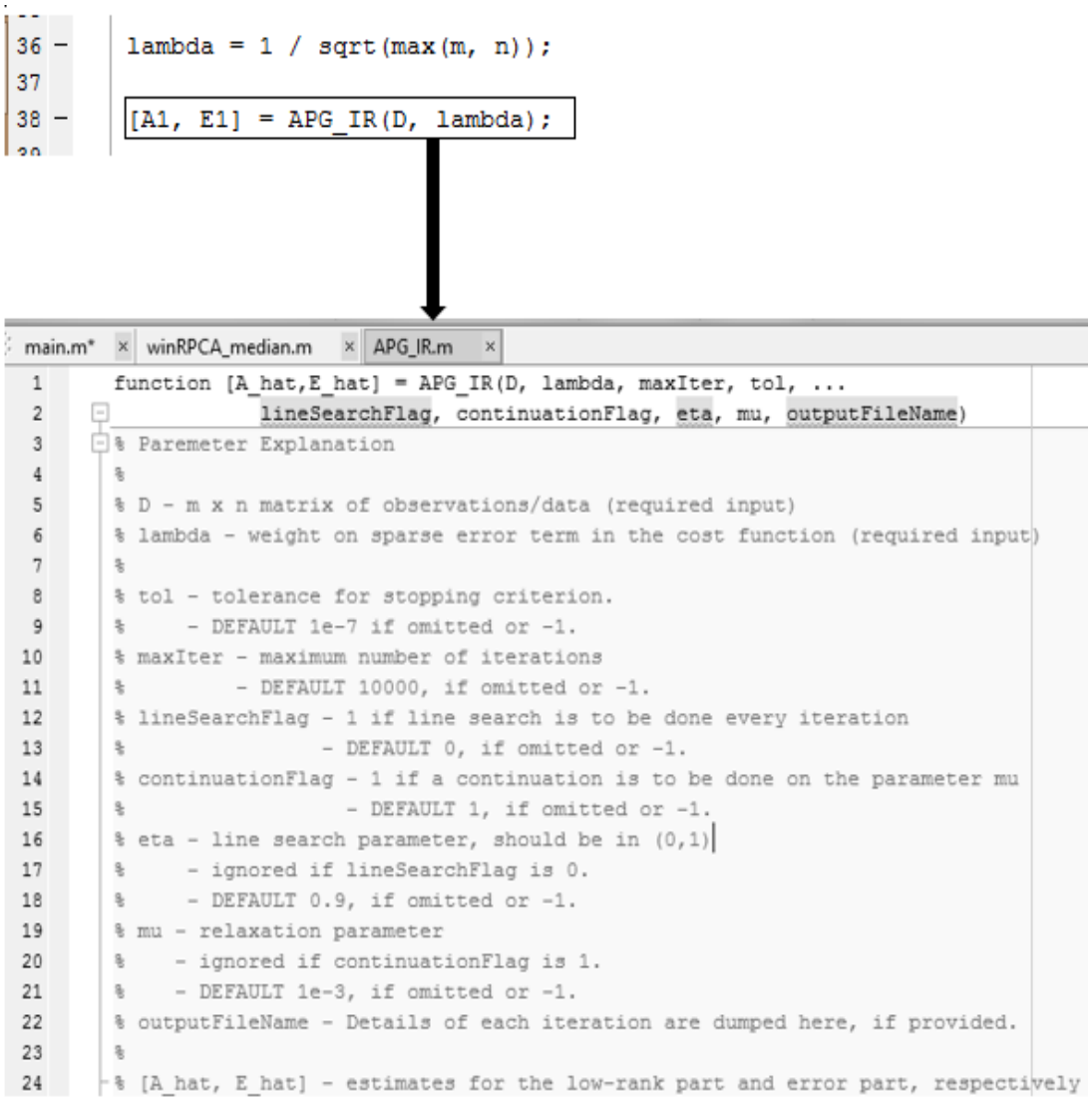
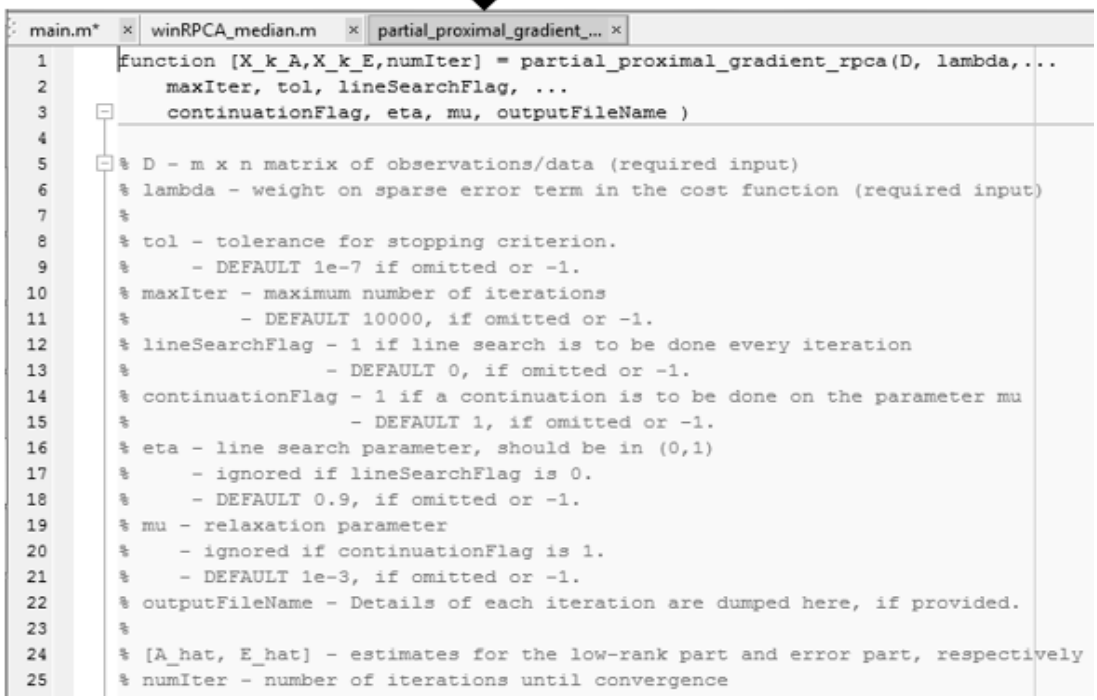


Figure 4.7. Calling APG from winRPCA_median and parameter explanation.

4.5.3 ACCELERATED PROXIMAL GRADIENT WITH PARTIAL SVD

```
34 -    lambda = 1 / sqrt(max(m, n));
35
36 -    [A1, E1] = partial_proximal_gradient_rpca(D, lambda);
```



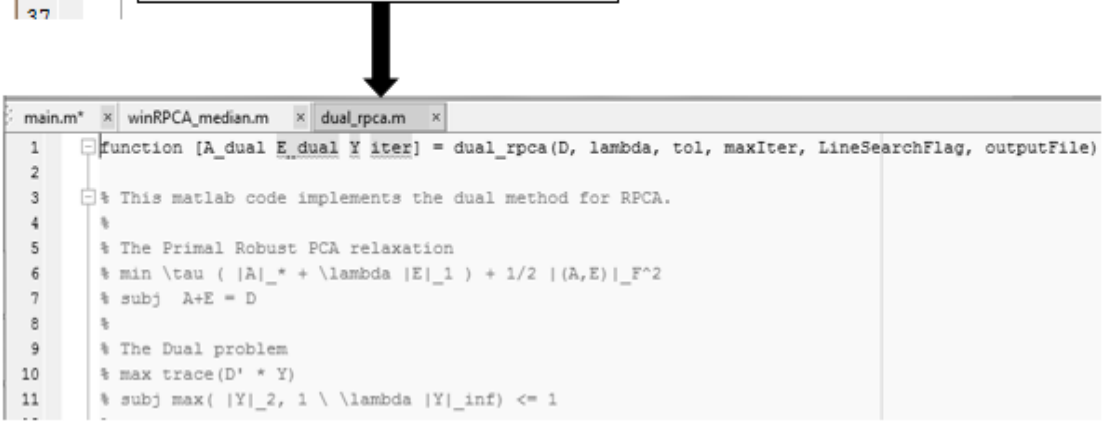
```
main.m* | winRPCA_median.m | partial_proximal_gradient_...

1   function [X_k_A,X_k_E,numIter] = partial_proximal_gradient_rpca(D, lambda,...
2       maxIter, tol, lineSearchFlag, ...
3       continuationFlag, eta, mu, outputFileName )
4
5   % D - m x n matrix of observations/data (required input)
6   % lambda - weight on sparse error term in the cost function (required input)
7   %
8   % tol - tolerance for stopping criterion.
9   %     - DEFAULT 1e-7 if omitted or -1.
10  % maxIter - maximum number of iterations
11  %         - DEFAULT 10000, if omitted or -1.
12  % lineSearchFlag - 1 if line search is to be done every iteration
13  %                - DEFAULT 0, if omitted or -1.
14  % continuationFlag - 1 if a continuation is to be done on the parameter mu
15  %                  - DEFAULT 1, if omitted or -1.
16  % eta - line search parameter, should be in (0,1)
17  %     - ignored if lineSearchFlag is 0.
18  %     - DEFAULT 0.9, if omitted or -1.
19  % mu - relaxation parameter
20  %    - ignored if continuationFlag is 1.
21  %    - DEFAULT 1e-3, if omitted or -1.
22  % outputFileName - Details of each iteration are dumped here, if provided.
23  %
24  % [A_hat, E_hat] - estimates for the low-rank part and error part, respectively
25  % numIter - number of iterations until convergence
```

Figure 4.8. Calling partial_proximity_gradient from winRPCA_median and parameter explanation.

4.5.4 DUAL METHOD

```
34 -    lambda = 1 / sqrt(max(m, n));
35
36 -    [A1, E1] = dual_rpca(D, lambda);
```



```
main.m* | winRPCA_median.m | dual_rpca.m

1   function [A_dual E_dual Y iter] = dual_rpca(D, lambda, tol, maxIter, LineSearchFlag, outputFile)
2
3   % This matlab code implements the dual method for RPCA.
4   %
5   % The Primal Robust PCA relaxation
6   % min \tau ( |A|_* + \lambda |E|_1 ) + 1/2 |(A,E)|_F^2
7   % subj  A+E = D
8   %
9   % The Dual problem
10  % max trace(D' * Y)
11  % subj max( |Y|_2, 1 \ \lambda |Y|_inf) <= 1
```

Figure 4.9. Calling dual_rpca from winRPCA_median and parameter explanation.

4.5.5 EXACT AUGMENTED LAGRANGE MULTIPLIER

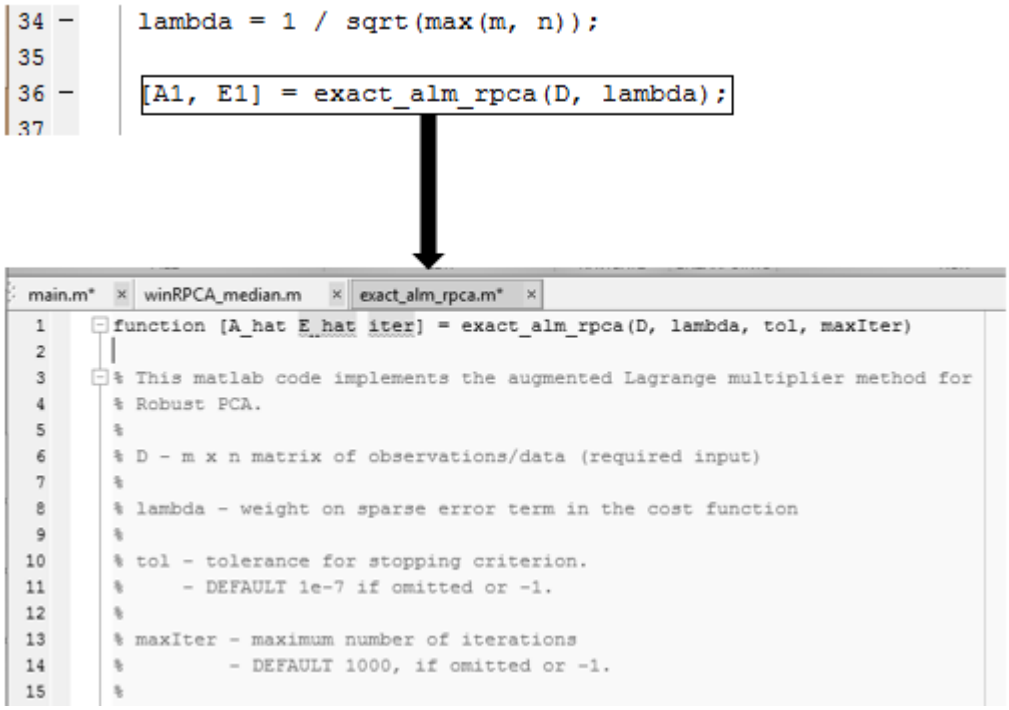


Figure 4.10. Calling exact_alm_rpca from winRPCA_median and parameter explanation.

4.5.6 INEXACT AUGMENTED LAGRANGE MULTIPLIER

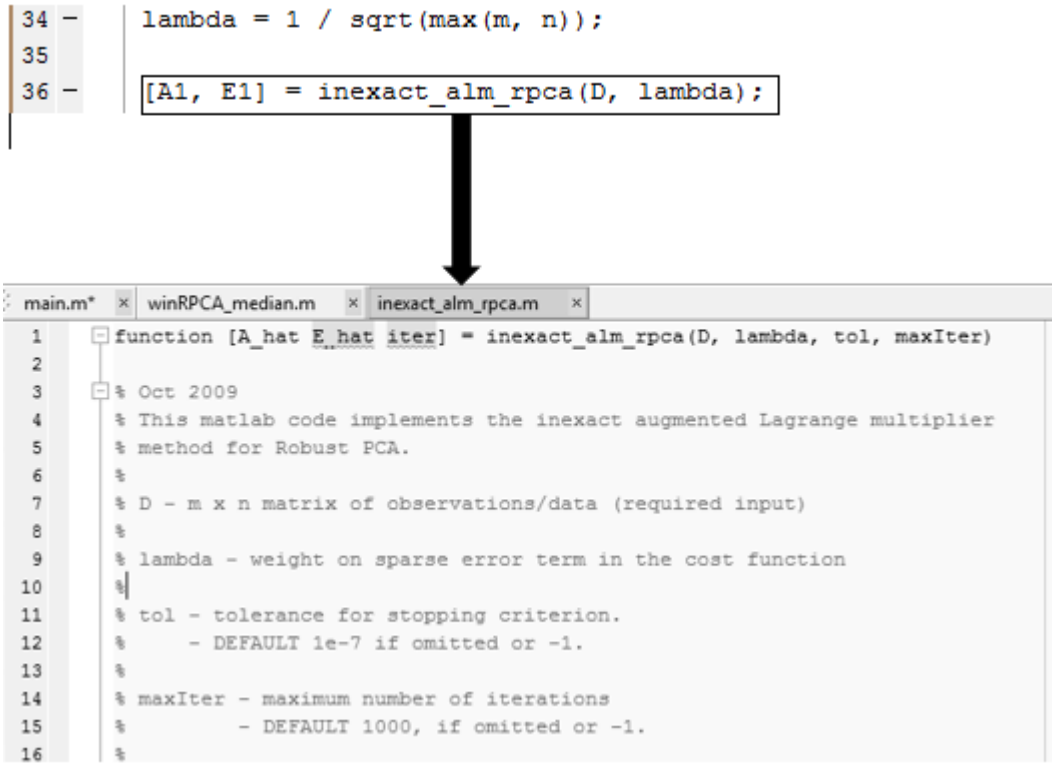


Figure 4.11. Calling inexact_alm_rpca from winRPCA_median and parameter explanation.

4.6 EVALUATION METRICS

In this section, we firstly introduce the evaluation metrics and the baseline methods for comparison in this project. Then we perform simulation experiments to evaluate the effects of parameters of the proposed method and its performances with respect to target numbers and the window sizes.

4.6.1 METRICS AND BASELINE METHODS

The most important metrics of evaluating the detection performance are the probability of detection P_d and false alarm rate F_a which are defined as following [21]:

- *Probability of detection*

Probability of detection is the measure of chances of a target to get detected by the applied algorithm. It gives the measure of accuracy of the method used for the small target detection in an infrared image. Higher the Probability of detection better is the method being used for the detection of small target in the infrared image.

$$P_d = \frac{\text{\# number of true detections}}{\text{\# number of actual targets}},$$

- *False Alarm Rate*

False alarm rate gives the average number of false detections per image where false detections refers to the condition when a target was signalled by the program when there was no target present at that position. Lower the value of this metrics, better is the method used for target detection.

$$F_a = \frac{\text{\# number of false detections}}{\text{\# number of images}}.$$

The detected result is considered correct if it simultaneously meets two requirements:

(a) The result and a ground truth have overlap pixels.

(b) The pixel distance between centres of the ground truth and the result is within a threshold (4 pixels).

4.6.2 SYNTHETIC IMAGE GENERATION

The main requirement for the evaluation metrics is the knowledge of the ground truth but in case of real images we have not placed the targets ourselves so we cannot automate the system to define the evaluation metrics. Solution for this problem is the synthetic image generation thereby generating the experimental data by synthesizing the images. Since we have placed the targets ourselves so we know where in the image the targets are located and hence we have the ground truth data.

STEP 1: GENERATION OF TARGETS

In this step we will generate the targets from the available real targets using the bipolar interpolation to resize these four real targets, where the original sizes of these four real targets are 5 × 4, 8 × 4, 5 × 3 and 6 × 4, respectively. Suppose that the original size of a target is $m \times n$, we resize the target to $\alpha m \times \alpha n$, where:

$$\alpha \in \left(\frac{2}{\min(m,n)}, \frac{3}{\min(m,n)}, \ldots, \frac{i}{\min(m,n)}, \ldots, \frac{21}{\min(m,n)}\right)$$

In this way we can obtain 20 targets with different sizes for each original target. For example, the second target with size of 8 × 4 can produce 20 targets with sizes of 4 × 2, 6 × 3, 8 × 4, 10 × 5, . . ., 38 × 19, 40 × 20 and 42 × 21, respectively, where the target with size of 8 × 4 is the same as the original target. For convenience of synthesizing images below, the pixel values of all target images are normalized to (0,1).

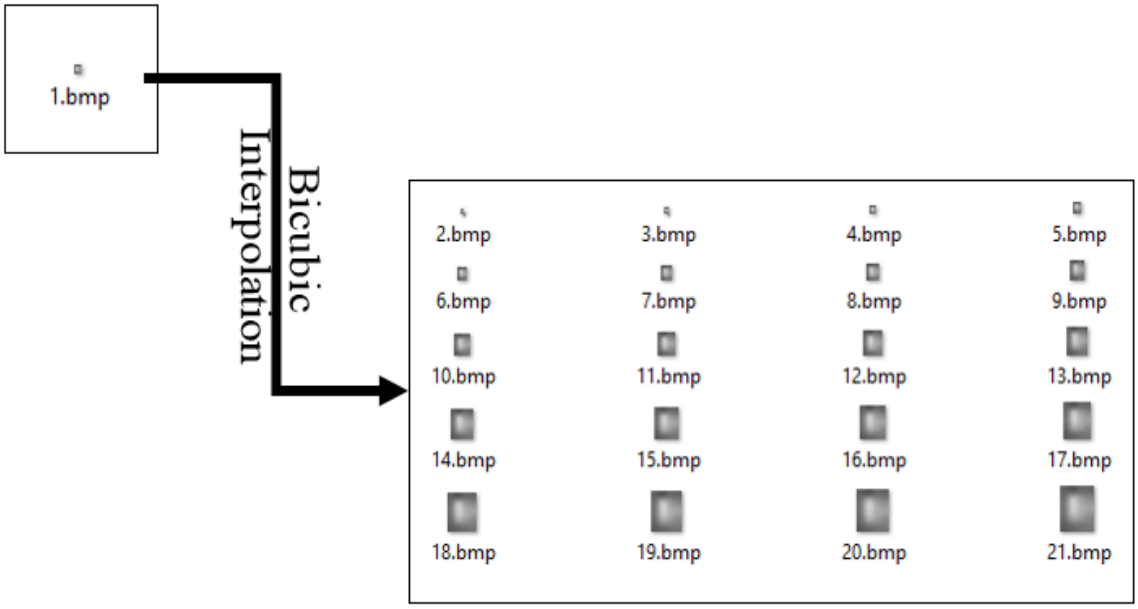


Figure 4.12. Bicubic interpolation of real targets.

STEP 2: EMBEDDING TARGETS ONTO BACKGROUND

A synthesized image f_D can be achieved by embedding a target image f_T with size of m x n into a background image f_B by the following way:

$$f_D(x,y) = \begin{cases} \max\Big(r f_T(x-x_0, y-y_0), \\ \quad f_B(x,y)\Big) & x \in (1+x_0, n+x_0), \\ & y \in (1+y_0, m+y_0) \\ \\ f_B(x,y) & \text{Otherwise,} \end{cases}$$

where (x_0, y_0) is a randomly produced pixel location which the left upper corner of the image f_T corresponds to in the image f_B, r is also produced randomly within the range of [h, 255], here h is the maximum pixel value of the background image patch covered by the image f_T. Finally we use a Gaussian filter to blur the synthesized target to make it close to a real one. This way, targets in the synthesized image are usually smaller than their original sizes. Actually, the same target usually have different sizes in different locations even in the same synthesized image. In the above simulation method, four image sequences are synthesized and all images in the same sequence have the same number of embedded targets which are 1, 4, 7 and 10, respectively.

While synthesizing the images we are placing the targets randomly so there is the chance that there might arise a situation that the multiple random generated pixel values might overlap sometimes. Sometimes though the randomly generated target pixel positions are not overlapping but they might be too close to be considered different targets and hence the performance calculation becomes ambiguous. To deal with this in every iteration when we generate a random pixel position if it is in 20 x 20 range of an already generated pixel, we discard this position and generate another position. This process is continued until we find a position which follows the above 20 x 20 rule.

Similarly to avoid a target position from getting a value which might result in the placement of the target not completely within the background image, we are discarding any position with less than 20 pixels distance from each side of the image. Thus, we are getting synthesized image with no overlapping targets and all the targets lying completely within the given background image.

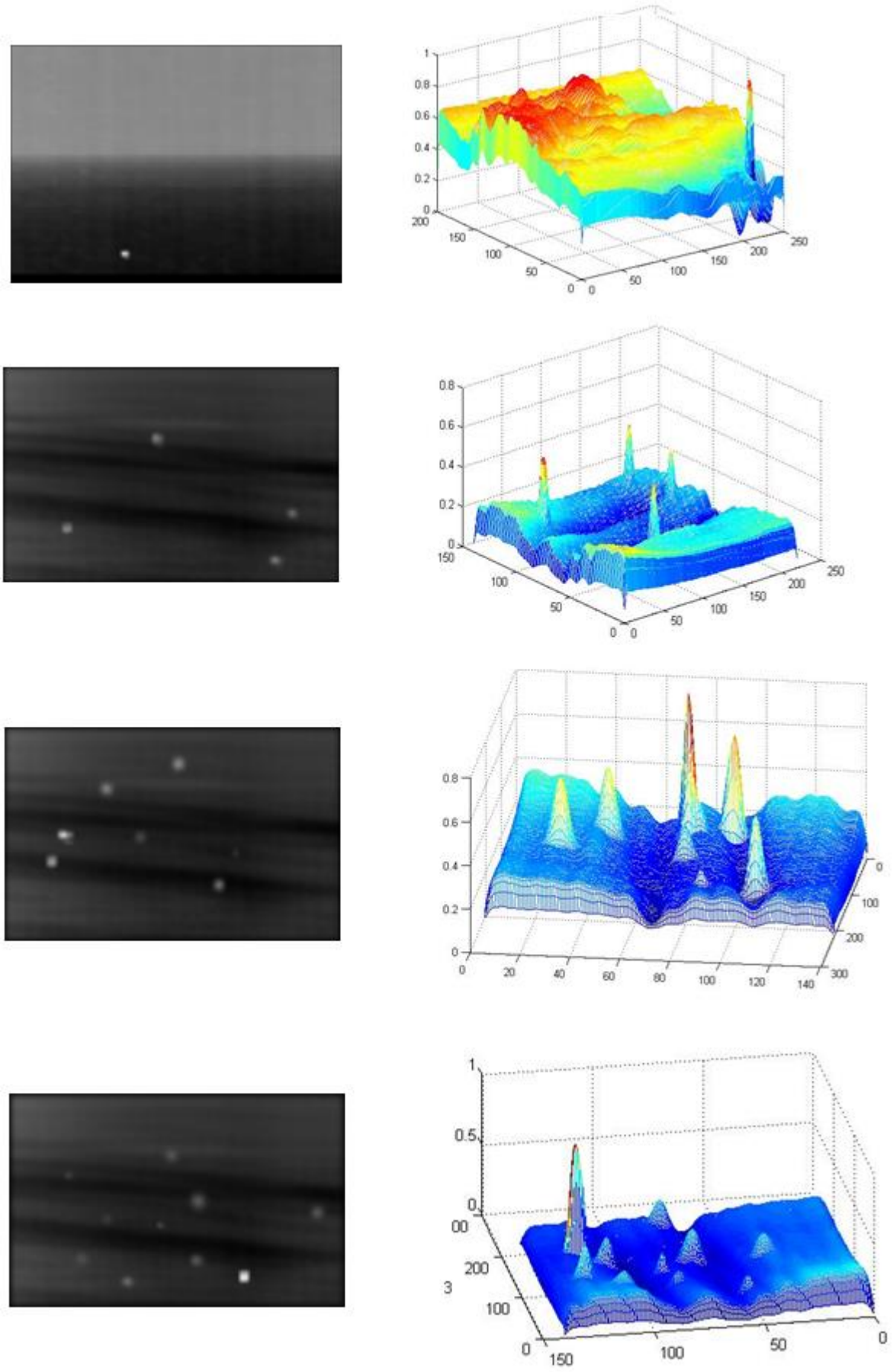


Figure 4.13. 1, 4, 7 and 10 targets synthetic image (left) and their 3D representation (right)

4.7 POST PROCESSING

By post-processing, we refine the segmentation result to obtain the final detection result. We are going to do some post processing to improve the results of the detection.

The steps done in post processing by us to get the refined result goes as follows:

(a) Suppresses structures that are lighter than their surroundings in the target image.

binaryImage = imclearborder(E);

(b) Find the image in which the connected components (detections) are clearly segmented.

LabeledImage = bwlabel (binaryImage);

(c) In this labeled image fill all the detections with size smaller than 4x4 pixels

BW = bwareaopen (labeledImage, 4);

(d) Find the final labeled image and the total number of detections.

[LabeledImage, numberOfDetections] = bwlabel (BW);

Figure 4.14. (leftmost) Target output image, (middle) after step (b) and (rightmost) final image after (c) and (d)

4.8 OVERALL PROCESSSING USING 3D REPRESENTATION OF IMAGES

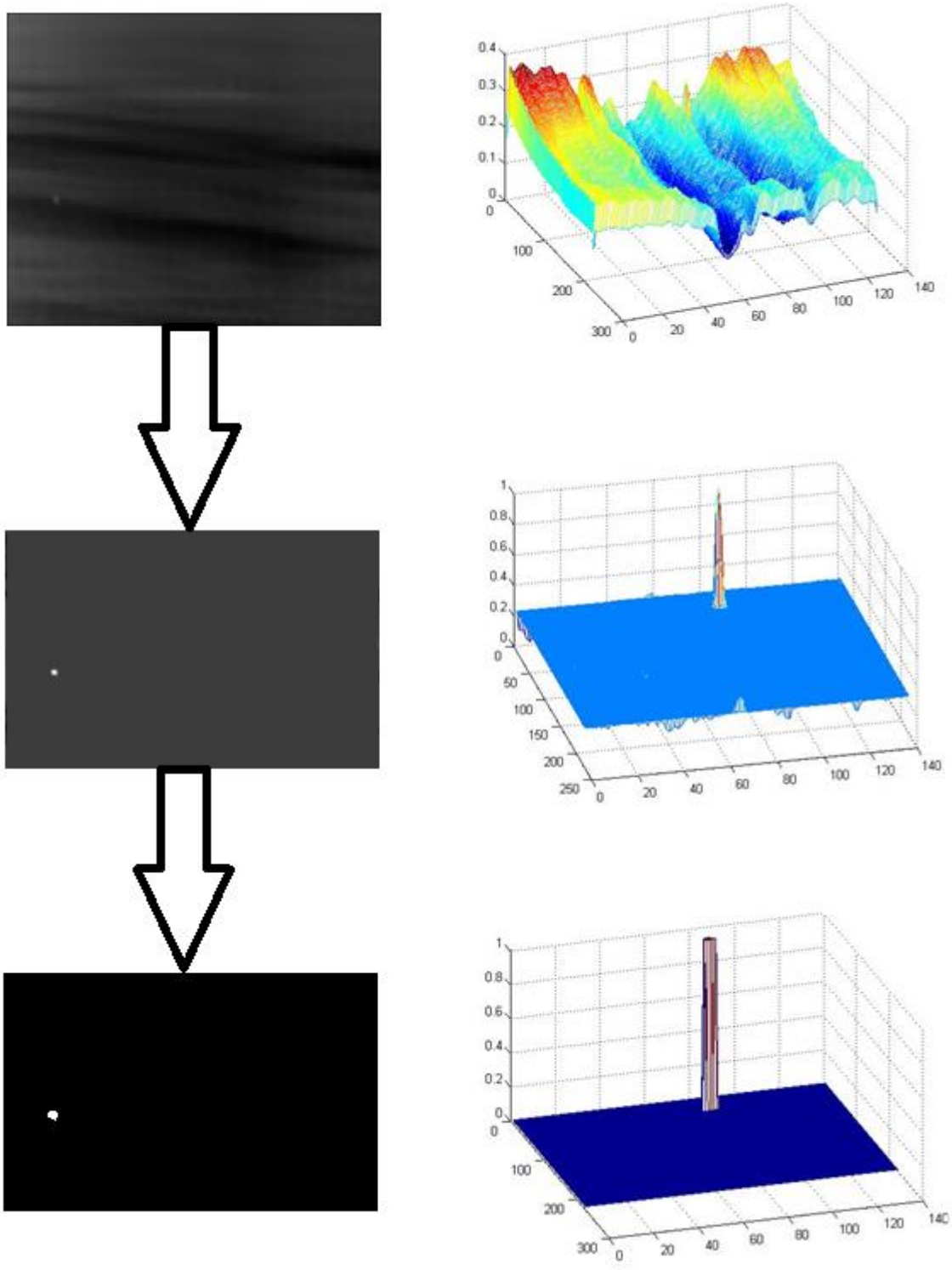


Figure 4.15. (left) Segmentation and post processing images and their corresponding 3D images (right).

CHAPTER 5
RESULTS AND DISCUSSION

5.1 REAL IMAGES: SEGMENTATION RESULT

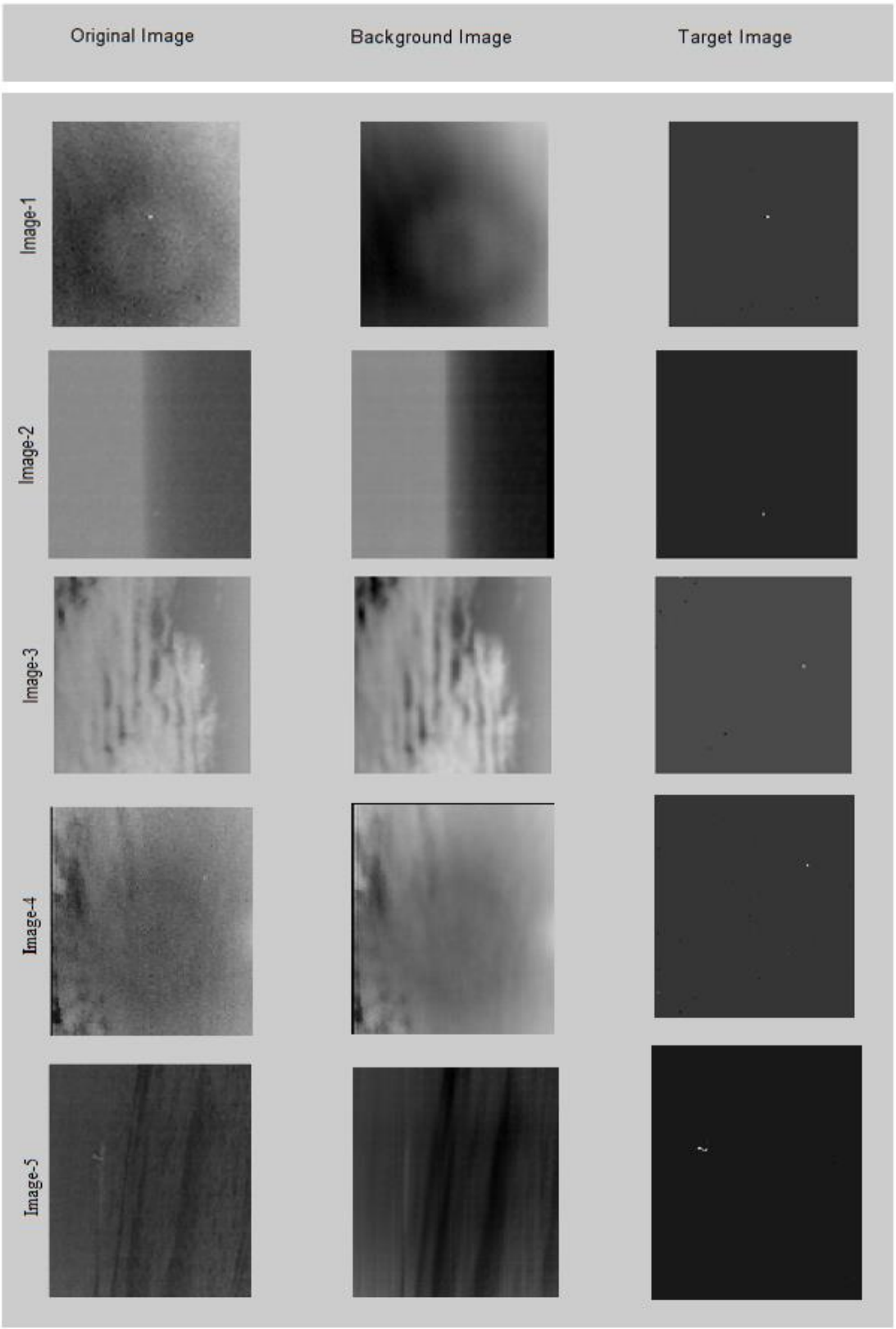


Figure 5.1. Real Images segmentation output

5.2 REAL IMAGES: TIME TAKEN FOR SEGMENTATION

Singular Value Thresholding

```
    Iteration 7800 rank(A) 174 ||E||_0 1243
    Iteration 7900 rank(A) 175 ||E||_0 1237
    Iteration 8000 rank(A) 177 ||E||_0 1236
    Iteration 8100 rank(A) 177 ||E||_0 1235
    Iteration 8200 rank(A) 177 ||E||_0 1239
    Iteration 8300 rank(A) 178 ||E||_0 1240
Elapsed time is 26178.867414 seconds.
>>
```

Accelerated Proximal Gradient

```
Command Window
  5/1: 1.bmp
  image/1.bmp
  5/2: 2.bmp
  image/2.bmp
  5/3: 3.bmp
  image/3.bmp
  5/4: 4.bmp
  image/4.bmp
  5/5: 5.bmp
  image/5.bmp
  Elapsed time is 70.825280 seconds.
>>
```

Accelerated Proximal Gradient (with partial SVD)

```
Iteration 60  rank(A) 180 ||E||_0 1189  Stopping Criterion
Iteration 80  rank(A) 180 ||E||_0 1200  Stopping Criterion
Iteration 100  rank(A) 180 ||E||_0 1200  Stopping Criterion
Iteration 120  rank(A) 180 ||E||_0 1199  Stopping Criterion
Elapsed time is 195.830026 seconds.
>>
```

Dual Method

```
970 Y_principal 47 |E|_0 641 objvalue 793.8358 |D-A-E|_F 29
980 Y_principal 40 |E|_0 649 objvalue 794.8751 |D-A-E|_F 31
990 Y_principal 51 |E|_0 555 objvalue 795.4335 |D-A-E|_F 26
1000 Y_principal 52 |E|_0 659 objvalue 796.3548 |D-A-E|_F 2
Maximum iterations reached
Elapsed time is 2577.495739 seconds.
>>
```

Exact Augmented Lagrange Multiplier

```
Iteration3 #svd 6 r(A) 2 |E|_0 0 stopCriterion 0.24019
Iteration4 #svd 12 r(A) 19 |E|_0 210 stopCriterion 0.14546
Iteration5 #svd 45 r(A) 180 |E|_0 1285 stopCriterion 0.0251
Iteration6 #svd 154 r(A) 180 |E|_0 1286 stopCriterion 3.865
Iteration7 #svd 156 r(A) 180 |E|_0 1286 stopCriterion 2.232
Elapsed time is 261.129329 seconds.
>>
```

Inexact Augmented Lagrange Multiplier

```
#svd 10 r(A) 139 |E|_0 2942 stopCriterion 0.049827
#svd 20 r(A) 180 |E|_0 1445 stopCriterion 8.0333e-07
Elapsed time is 69.186632 seconds.
>>
```

Figure 5.2. Time calculation instances from MATLAB

- Time is taken in seconds approximated to the nearest integer.
- All solutions were carried out on Intel(R) Core(TM) i5-3230M CPU @2.60 GHz CPU , 4 GB RAM, 64-bit Microsoft Windows 8 operating system.

Algorithm	Time (s)
Singular Value Thresholding	26179
Accelerated Proximal Gradient	71
Accelerated Proximal Gradient (with partial SVDs)	196
Dual Method	2577
Exact ALM	248
Inexact ALM	69

Table 5.1. Time comparison between different convex optimization algorithms.

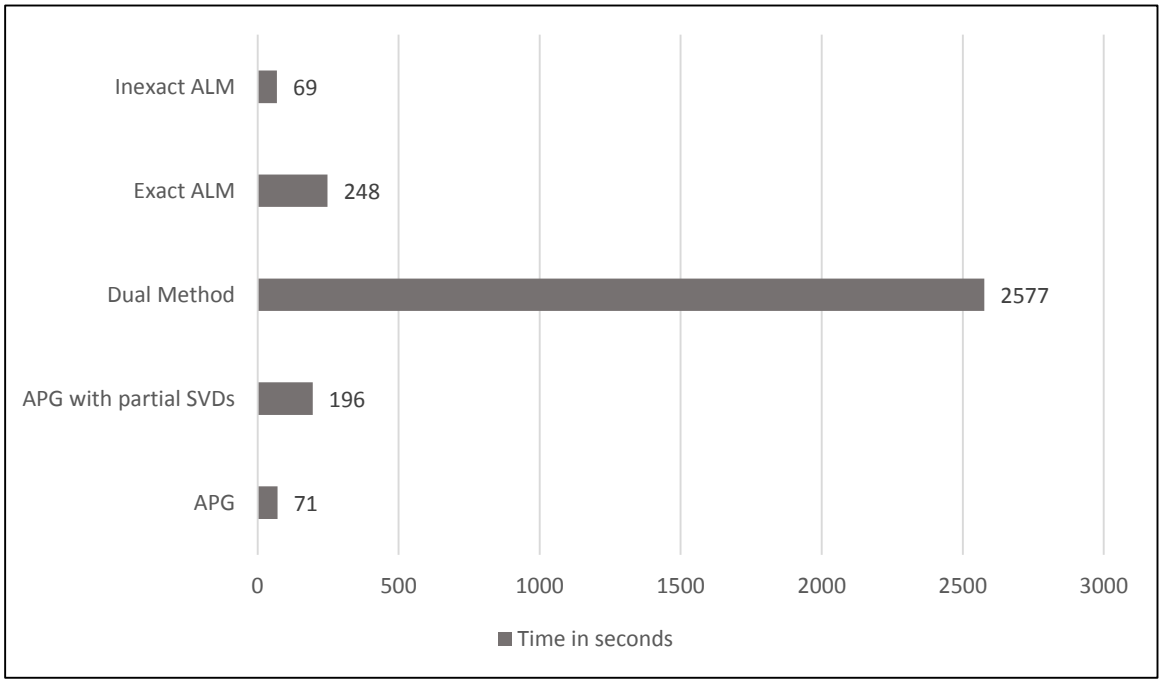


Figure 5.3. Comparison between different algorithms

5.3 EXPERIMENTAL DATA: COMPARISON BETWEEN ALGORITHMS

On the basis of running time calculated on the real images two algorithms stand separately from rest of them APG and Inexact ALM. Thus for the further calculation on the synthetic images for the evaluation of performance metrics we have used only these two algorithms.

Moreover in the third method we are combining the segmentation result of APG and Inexact ALM before post processing to form a new segmentation result. This result gives good result with single target but as the number of targets increase in the images its performance has shown exponential downfall, hence resulting in the degradation with respect to the increasing number of targets.

Thus we are showing all the results with respect to only these three algorithms because of their good results and fast processing while detecting target in the corresponding real image segmentation experiment. So these three algorithms are:

- Accelerated Proximal Gradient
- Inexact Augmented Lagrange Multiplier
- APG + Inexact ALM

RESULTS:

Algorithm	Time (s)	Number of targets	Probability of Detection (Average)	False Alarm Rate(per image)
Accelerated Proximal Gradient	1782.911024	1	0.7900	0.9000
Inexact ALM	1114.942697	1	0.8400	1.2200
Inexact ALM + APG	2897.853721	1	0.8400	0.5600

Table 5.2. Performance Comparison on images with single target.

Algorithm	Time (s)	Number of targets	Probability of Detection (Average)	False Alarm Rate(per image)
Accelerated Proximal Gradient	1782.920528	4	0.7225	0.8600
Inexact ALM	1314.587001	4	0.7950	0.8100
Inexact ALM + APG	3097.507529	4	0.7825	0.8700

Table 5.3. Performance Comparison on images with 2 targets.

Algorithm	Time (s)	Number of targets	Probability of Detection (Average)	False Alarm Rate(per image)
Accelerated Proximal Gradient	1782.911024	7	0.6386	1.0600
Inexact ALM	1114.942697	7	0.7386	1.1500
Inexact ALM + APG	2897.853721	7	0.7257	1.9200

Table 5.4. Performance Comparison on images with 7 targets.

Algorithm	Time (s)	Number of targets	Probability of Detection (Average)	False Alarm Rate(per image)
Accelerated Proximal Gradient	1772.570794	10	0.5870	1.2500
Inexact ALM	1126.028480	10	0.7010	1.2900
Inexact ALM + APG	2898.599274	10	0.6850	3.1500

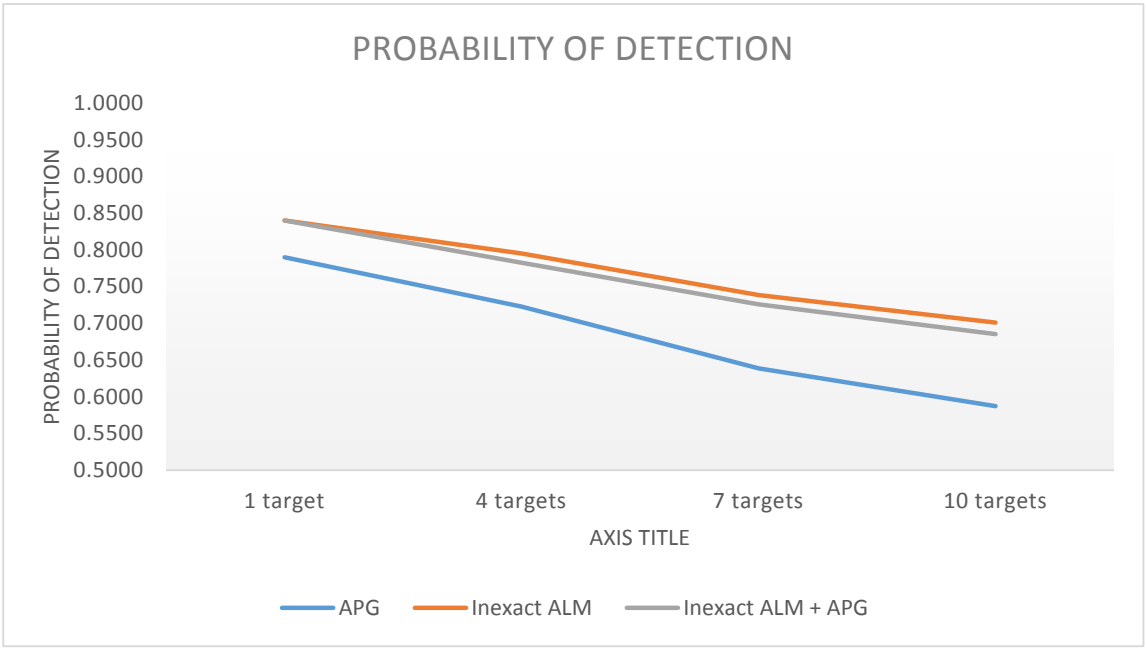
Table 5.5. Performance Comparison on images with 10 targets.

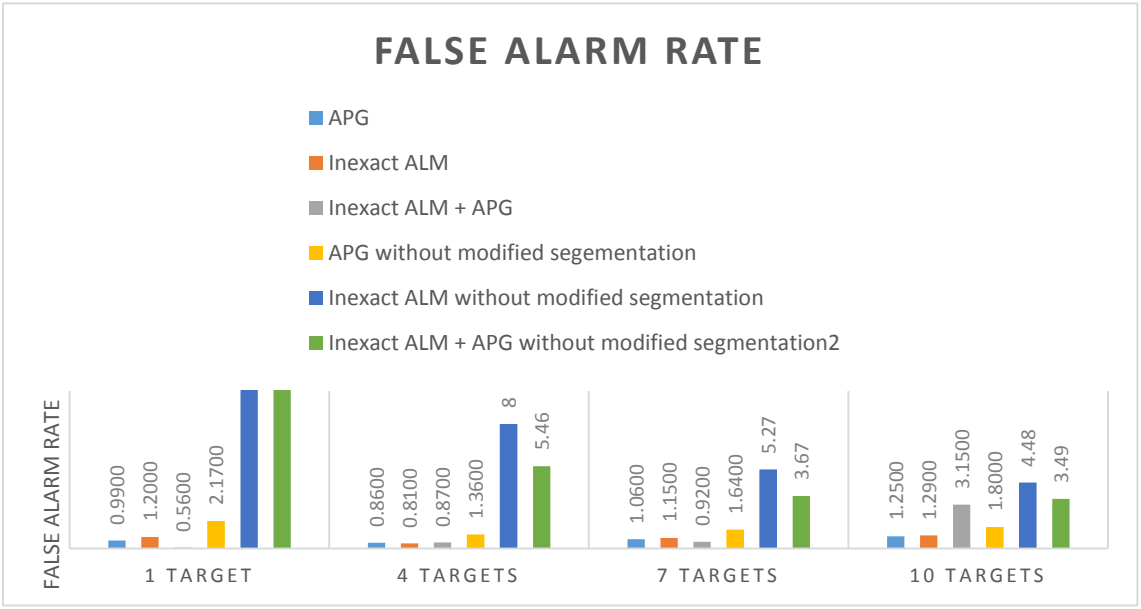
5.4 EXPERIMENTAL DATA: COMPARISON ON THE BASIS OF MODIFIED POST PROCESSING

In this table, Mod represents the result when the post processing neglects the detections smaller than 4 x 4 in size.

Algorithm	Time (s)	Number of targets	Probability of Detection (Average)		False Alarm Rate(per image)	
			Mod.	No Mod.	Mod.	No Mod.
Accelerated Proximal Gradient	1782.911024	1	0.7900	0.7900	0.9000	2.1700
Inexact ALM	1114.942697	1	0.8400	0.8500	1.2200	17.900
Inexact ALM + APG	2897.853721	1	0.8400	0.8500	0.5600	13.870 0
Accelerated Proximal Gradient	1782.920528	4	0.7225	0.7275	0.8600	1.3600
Inexact ALM	1314.587001	4	0.7950	0.8050	0.8100	8.0000
Inexact ALM + APG	3097.507529	4	0.7825	0.7925	0.8700	5.4600
Accelerated Proximal Gradient	1787.155656	7	0.6386	0.6514	1.0600	1.6400
Inexact ALM	1151.564336	7	0.7386	0.7629	1.1500	5.2700
Inexact ALM + APG	2938.719992	7	0.7257	0.7471	1.9200	3.6700
Accelerated Proximal Gradient	1772.570794	10	0.5870	0.6030	1.2500	1.8000
Inexact ALM	1126.028480	10	0.7010	0.7290	1.2900	4.4800
Inexact ALM + APG	2898.599274	10	0.6850	0.7100	3.1500	3.4900

Table 5.6. Performance Comparison on the basis of modified post processing.





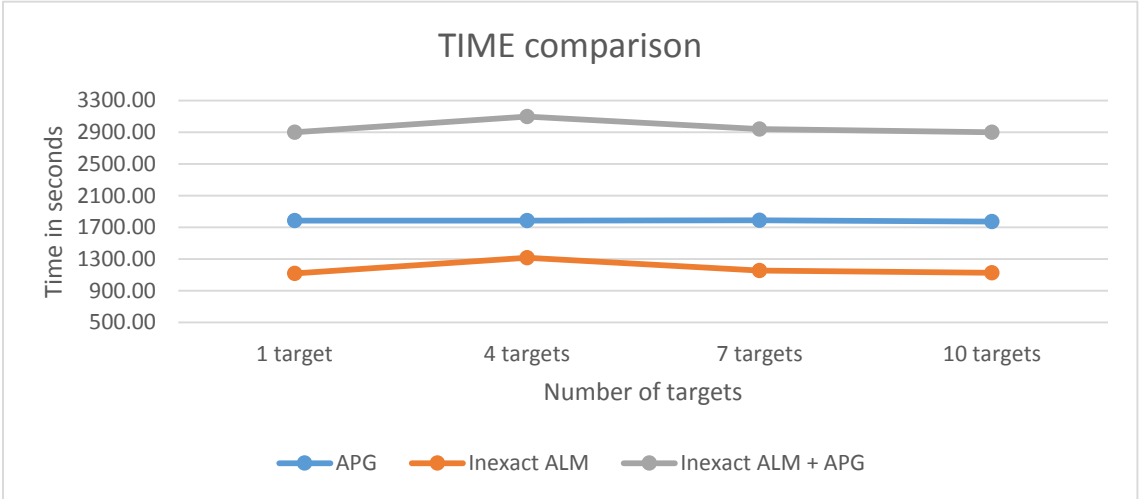


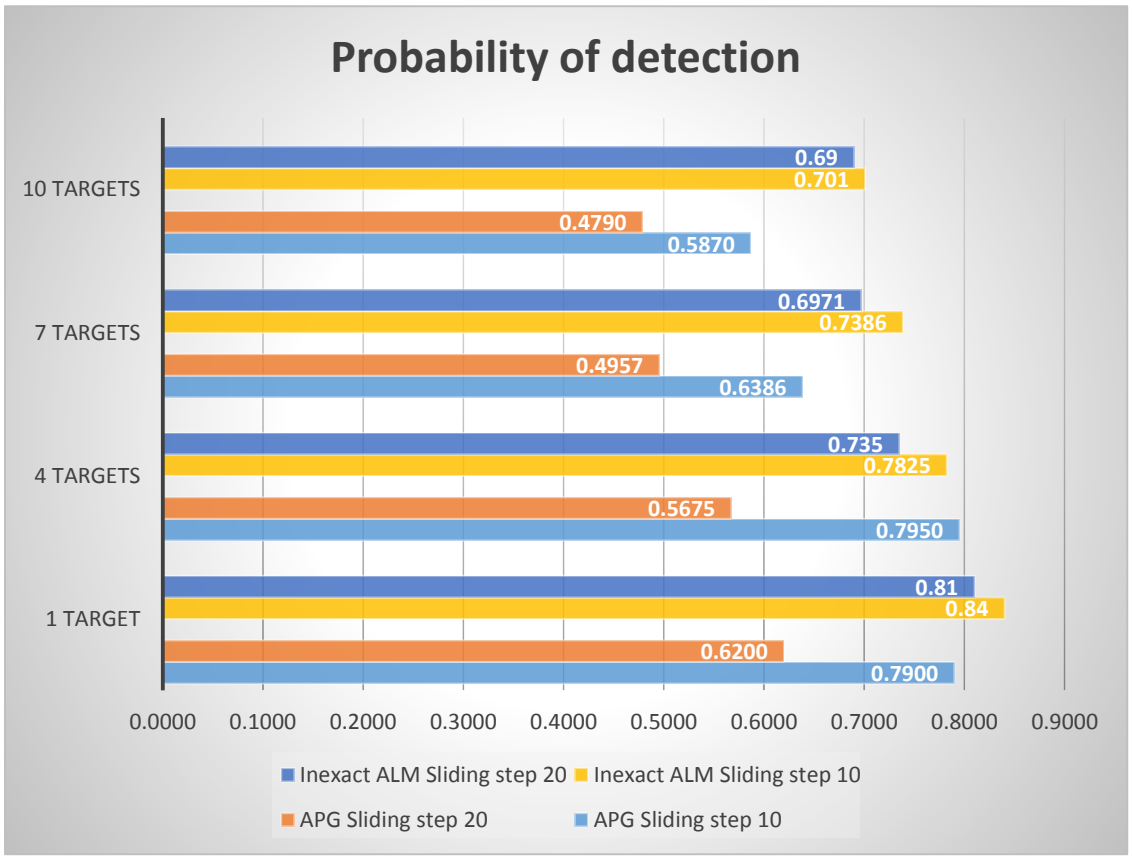
Figure 5.4. Performance Comparison between different algorithms

5.5 EXPERIMENTAL DATA: COMPARISON ON THE BASIS OF SLIDING STEP

In this section we are comparing the result of these three algorithms on the basis of sliding steps. SS in the table denotes the sliding step. We are focusing on the least time taken to detect the target so any reduction in the sliding step below 10 will lead to very high redundancy in the patch data values and any increase in the sliding step will lead to decrease in the performance of the algorithm. Anyway the Inexact ALM always supersedes the APG and the combined algorithm in case of time and probability of detection.

Algorithm	**Number of targets**	**Probability of Detection (Average)**		**False Alarm Rate(per image)**	
-	**-**	**SS 10**	**SS 20**	**SS 10**	**SS 20**
Accelerated Proximal Gradient	1	0.7900	0.6200	0.9000	0.5500
Accelerated Proximal Gradient	4	0.7950	0.5675	0.8100	1.0000
Accelerated Proximal Gradient	7	0.6386	0.4957	1.0600	1.2700
Accelerated Proximal Gradient	10	0.5870	0.4790	1.2500	1.2600
Inexact ALM	1	0.8400	0.8100	1.2200	5.1600
Inexact ALM	4	0.7825	0.7350	0.8700	3.9500
Inexact ALM	7	0.7386	0.6971	1.1500	3.3400
Inexact ALM	10	0.7010	0.6900	1.2900	3.2000

Table 5.7. Performance Comparison on the basis of sliding step.



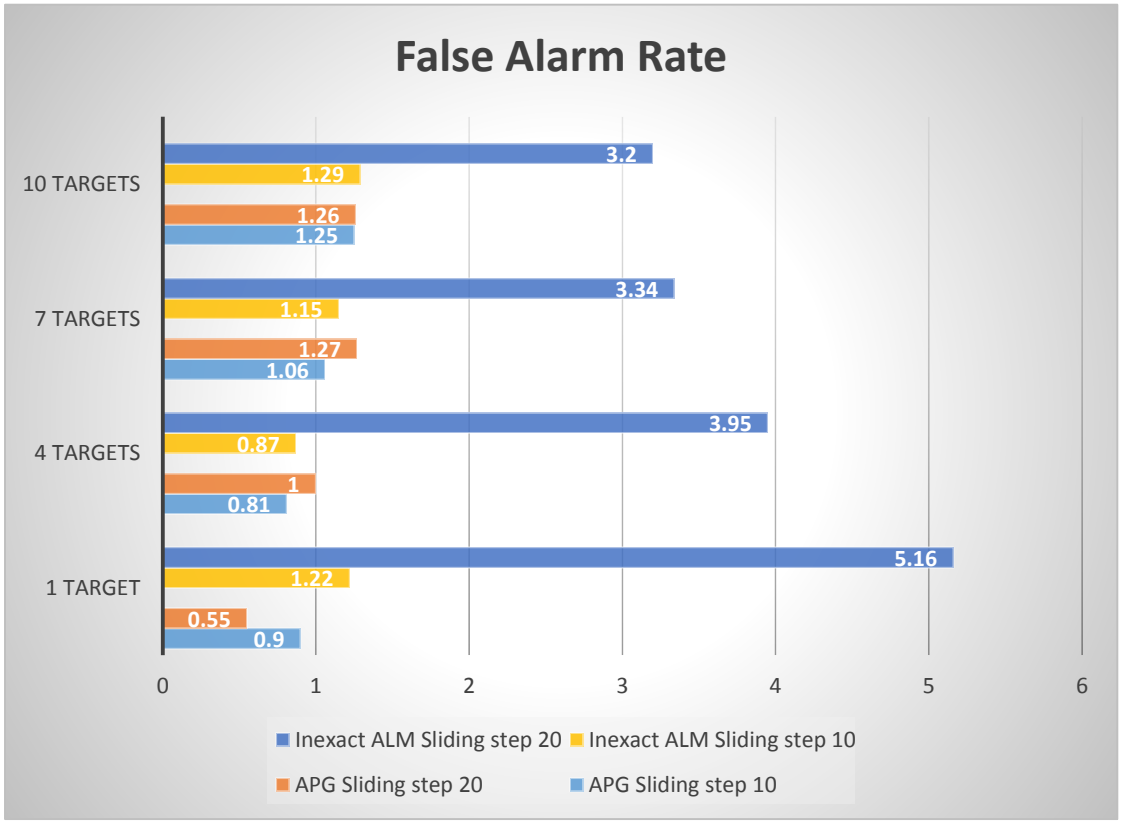


Figure 5.5. Performance Comparison on the basis of sliding steps

5.6 PROBABILITY OF DETECTION VS FALSE ALARM RATE

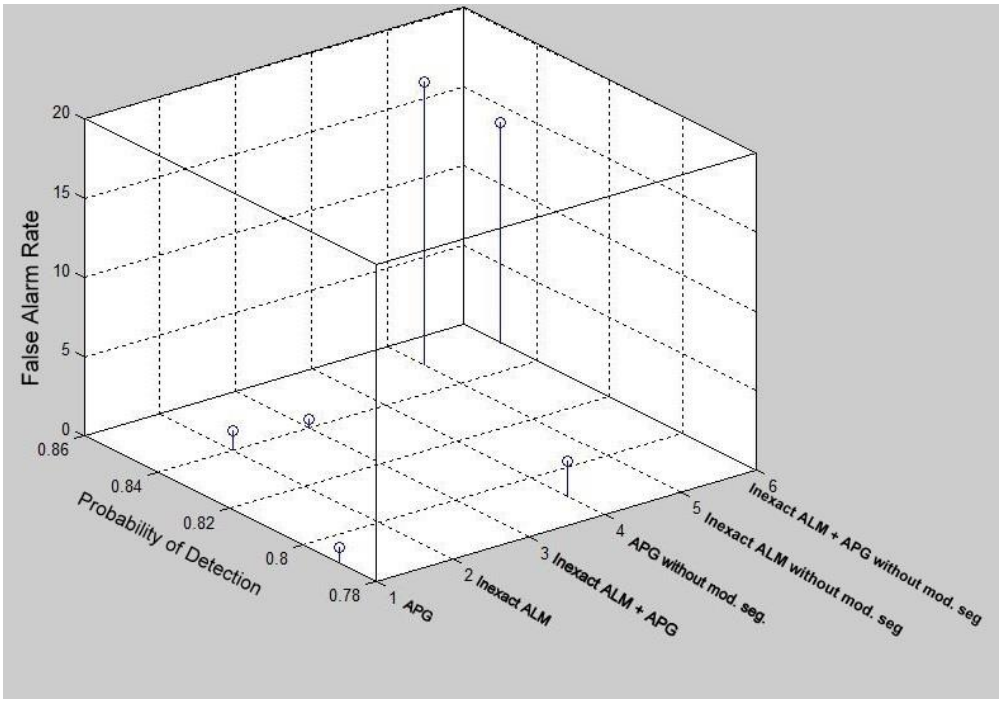


Figure 5.6. P_d vs F_a for images with 1 target

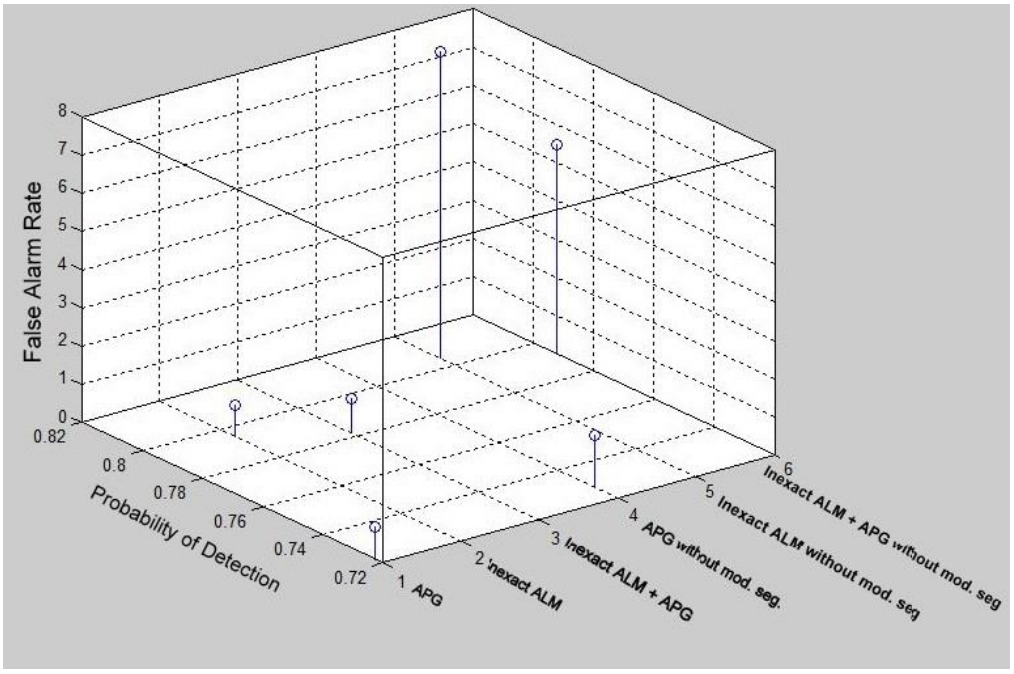


Figure 5.7. P_d vs F_a for images with 4 targets

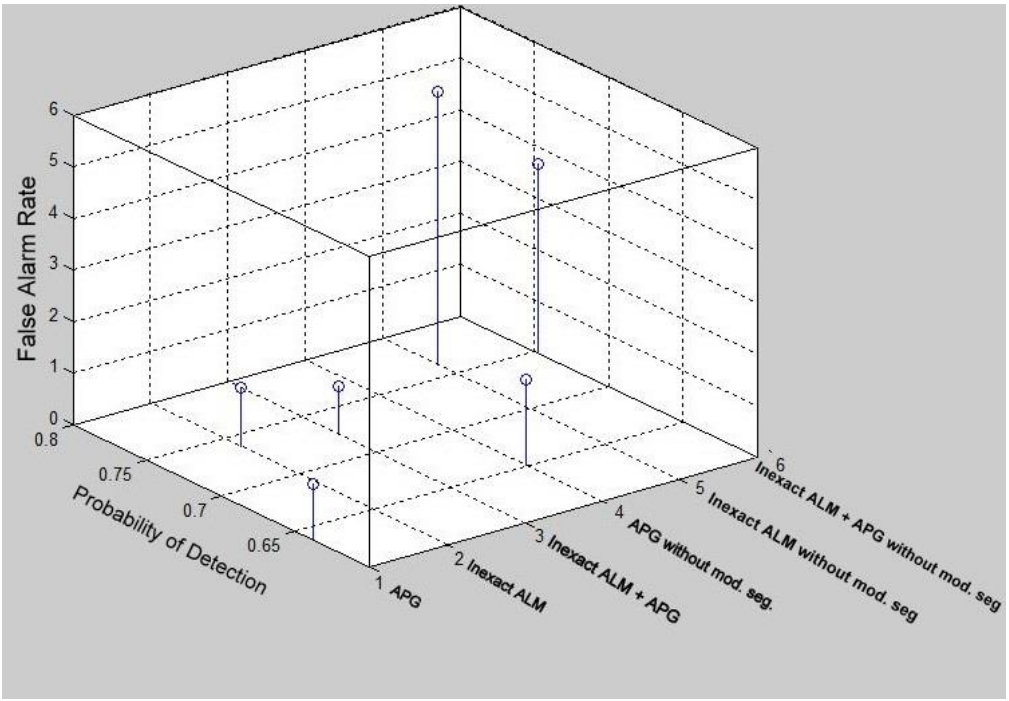


Figure 5.8. P_d vs F_a for images with 7 targets

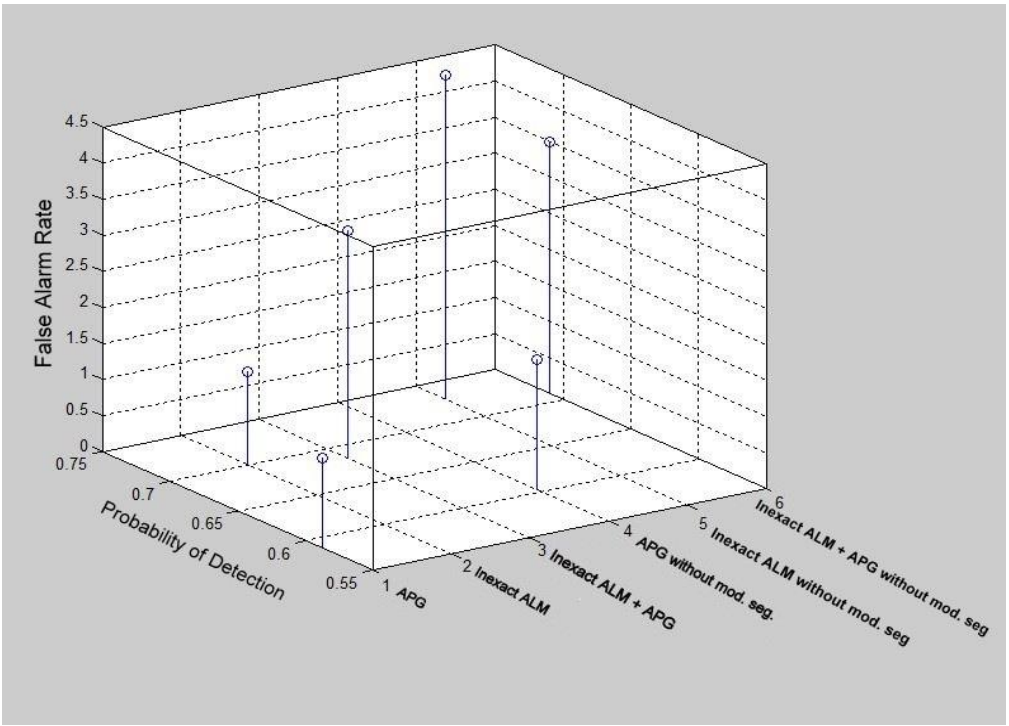


Figure 5.9. P_d vs F_a for images with 10 targets

5.7 DISCUSSION

The combination of Inexact ALM and APG gives exceptional results for single target image but as the number of targets increases its performance falls very fast and moreover the time taken by this algorithm is almost double than both of the APG and Inexact ALM.

As we can see from the above results, tables and graphs that Inexact Augmented Lagrange Multiplier inarguable provides better Probability of detection and takes much less time than rest of the algorithms keeping the False Alarm Rate similar to that of the Accelerated Proximal Gradient. The slight rise in the False Alarm Rate of the Inexact Augmented Lagrange Multiplier compared to the Accelerated Proximal Gradient is not an issue because the rise is within the acceptable range such that approximately the number of false alarms per image remains same. For example if the APG had False Alarm Rate 0.9000 and the Inexact ALM has False Alarm Rate as 1.0500, then both the algorithms have approximately 1 false alarm per image. Thus the tradeoff in case of Inexact ALM is within the acceptable range.

CHAPTER 6
CONCLUSIONS AND FUTURE SCOPE

11.1 CONCLUSIONS

A new infrared image model called IPI model for small target detection is presented based on the non-local self-correlation property of the infrared image in this project. Then the small target detection task is transformed into an optimization problem of recovering low-rank and sparse matrices, which can be effectively solved by using Stable Principal Component Pursuit.

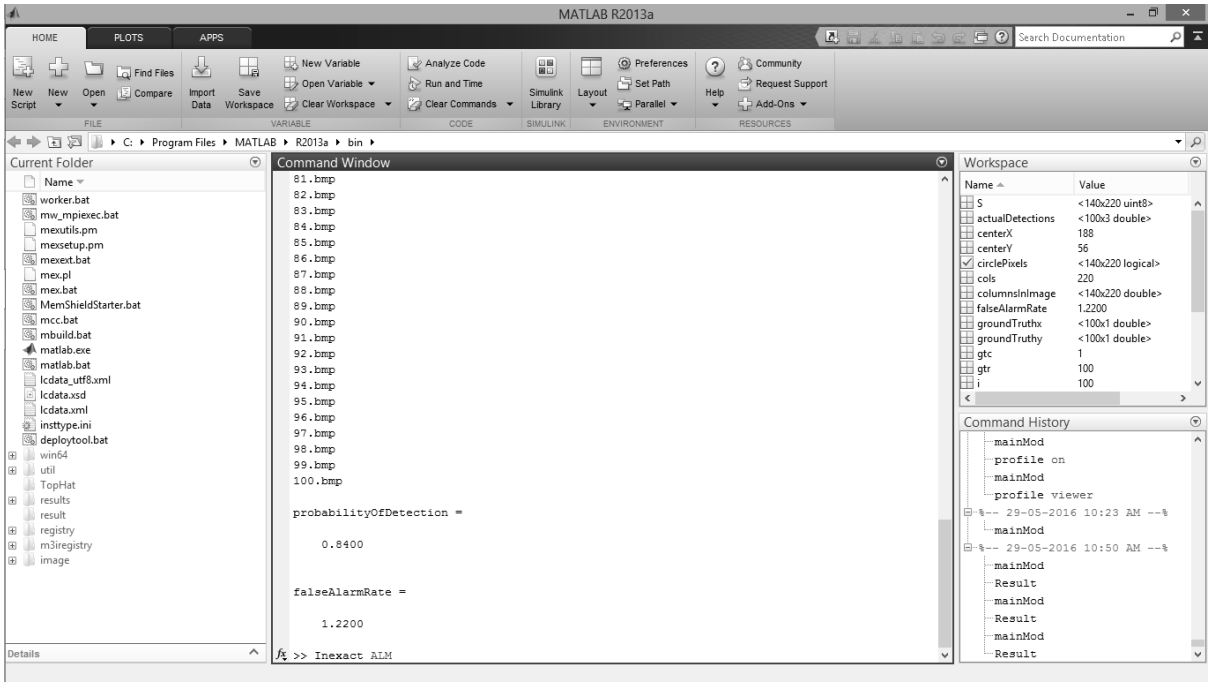
Though there is a slight rise in the False Alarm Rate of the Inexact Augmented Lagrange Multiplier with respect to the Accelerated Proximal Gradient but the rise is within the acceptable range as discussed in the Discussion section of the Chapter 5. But the tremendous rise in the performance in terms of time and Probability of detection has given a new real time application based challenge to the existing baseline methods.

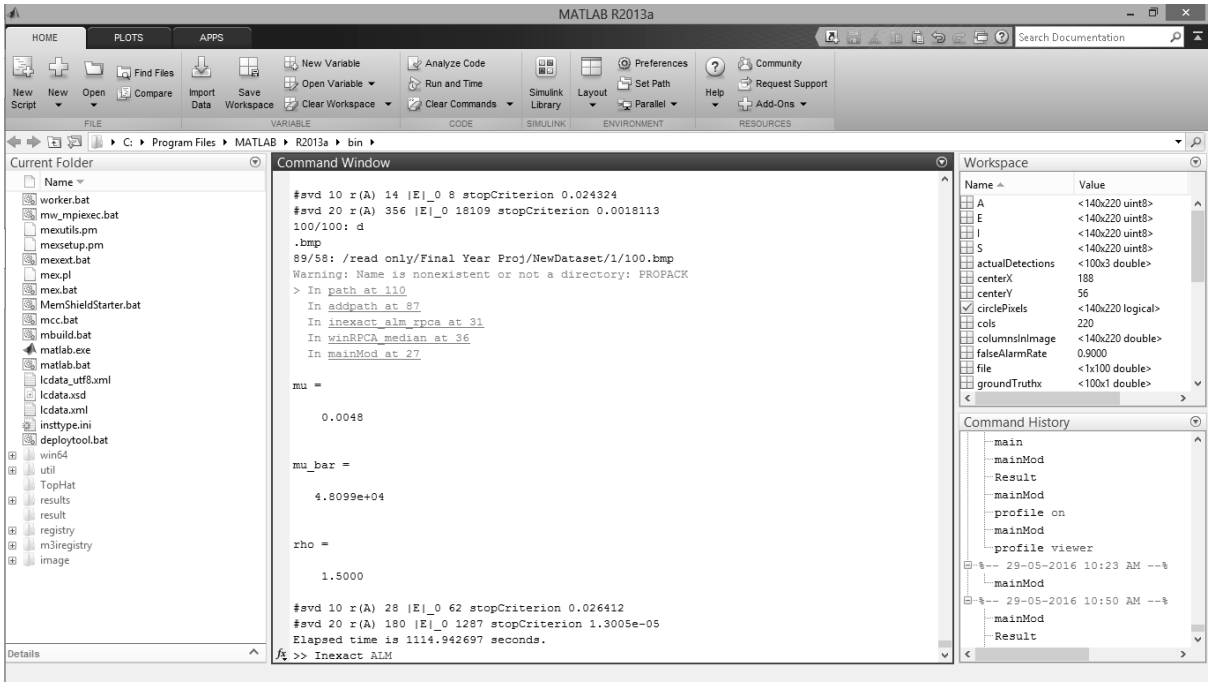
Thus from all the results and discussion we can conclude that Inexact Augmented Lagrange Multiplier is the one best served for the purpose of small target detection in infrared images and provides a higher Probability of detection and takes less time than the baseline method.

11.2 FUTURE SCOPE

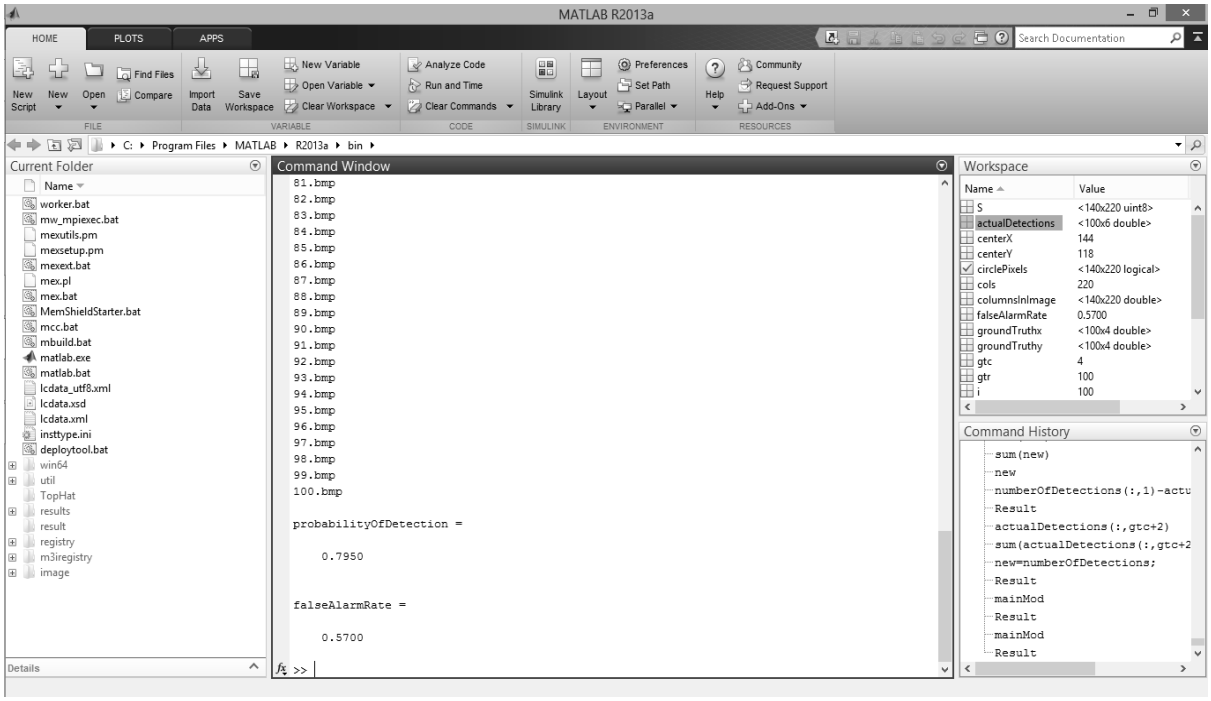
Extensive synthetic and real data experiments show that under different backgrounds the proposed method can not only work more stably for different number of targets, but also has better detection performance compared to conventional baseline methods. In the future, we will investigate faster version of the current algorithm. In addition, we will further generalize our 2D patch model into 3D or more dimensions and investigate applications of the N-D patch model. We will also try recent multi-subspace cluster strategies [42] to further improve the flexibility of our method in highly variant background cases in our future investigation. We will also work on improving the post processing to reduce the False Alarm Rate further.

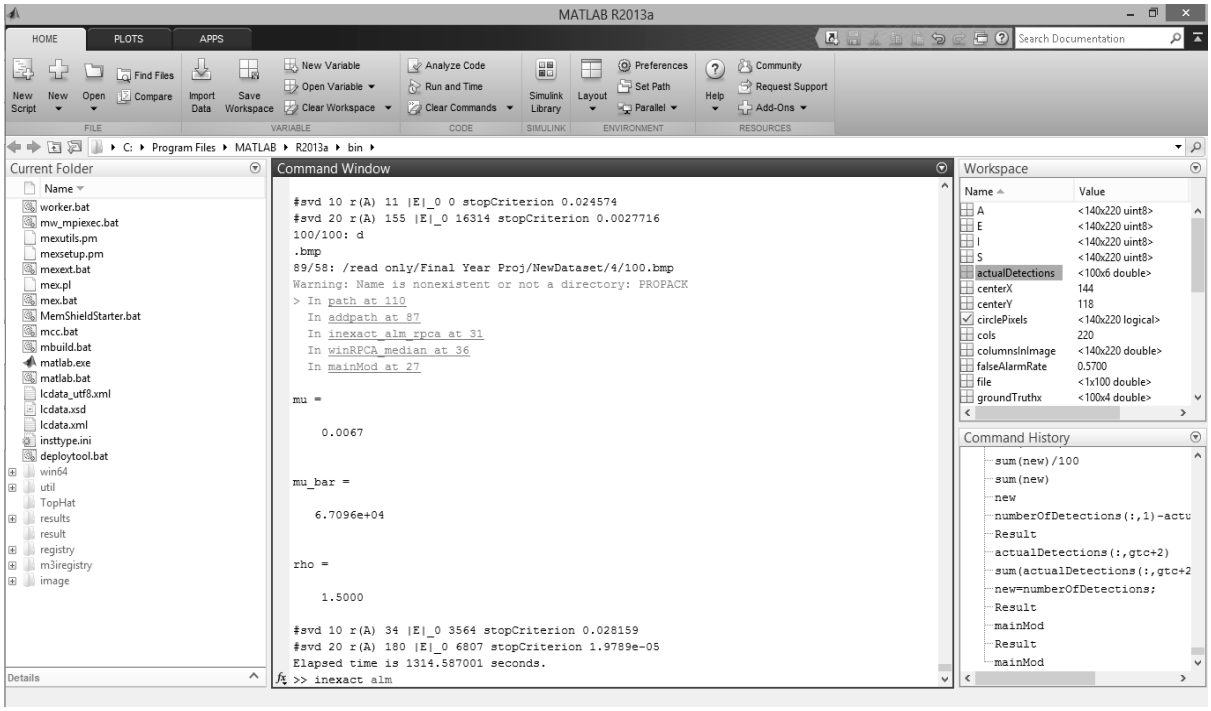
APPENDIX A: SCREENSHOTS



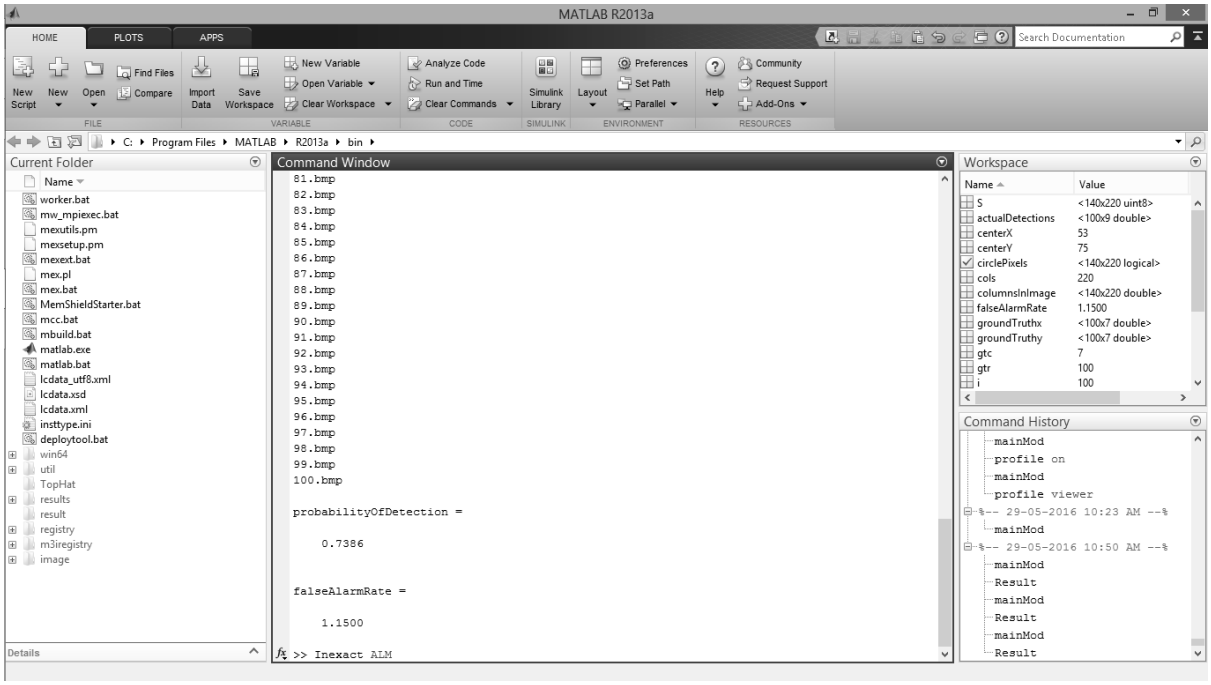


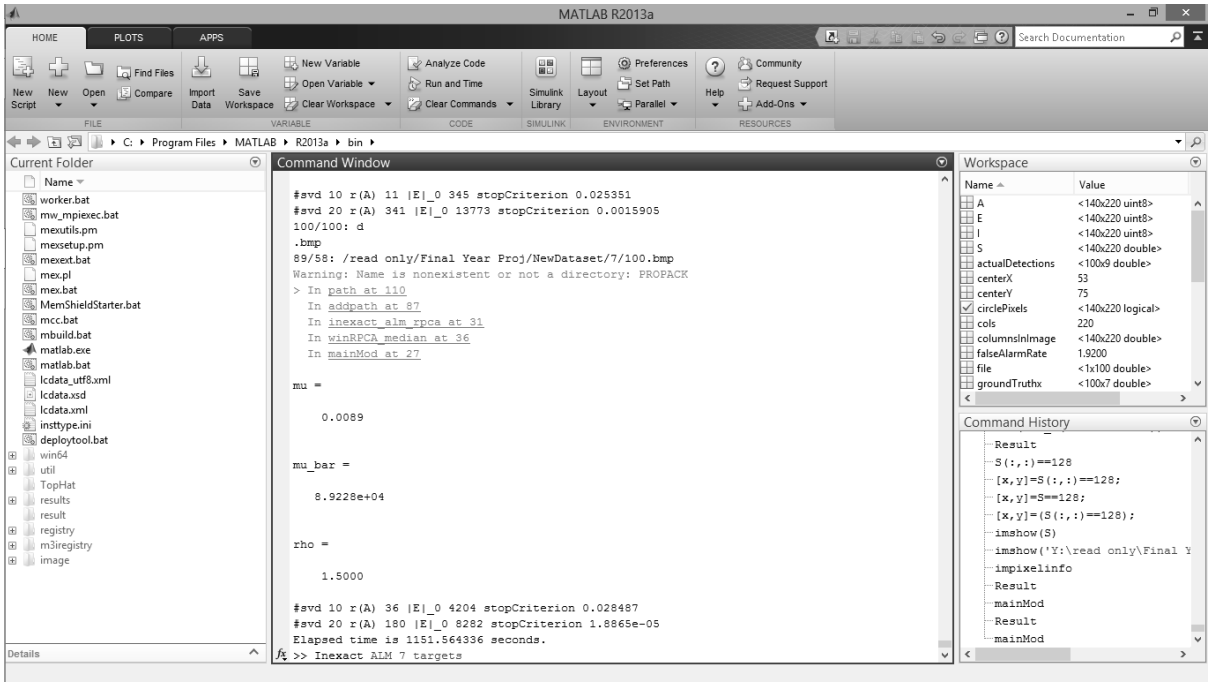
MATLAB Terminal Output (performance and time) after running Inexact ALM on 100 images with 1 target



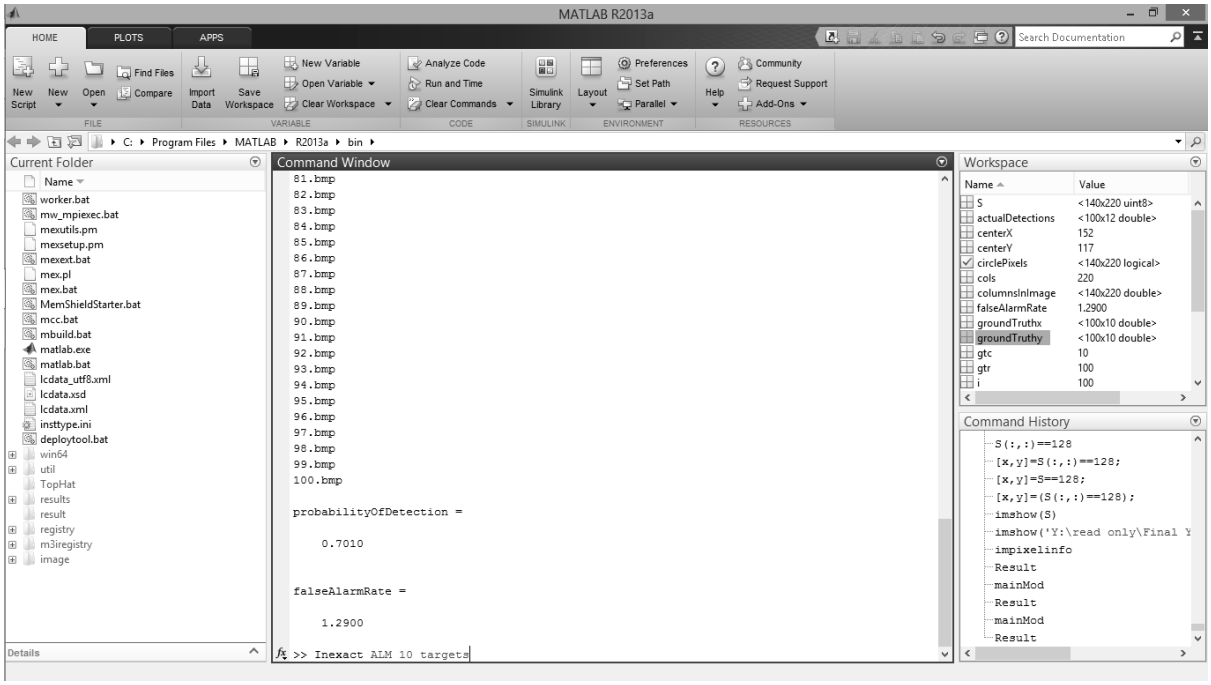


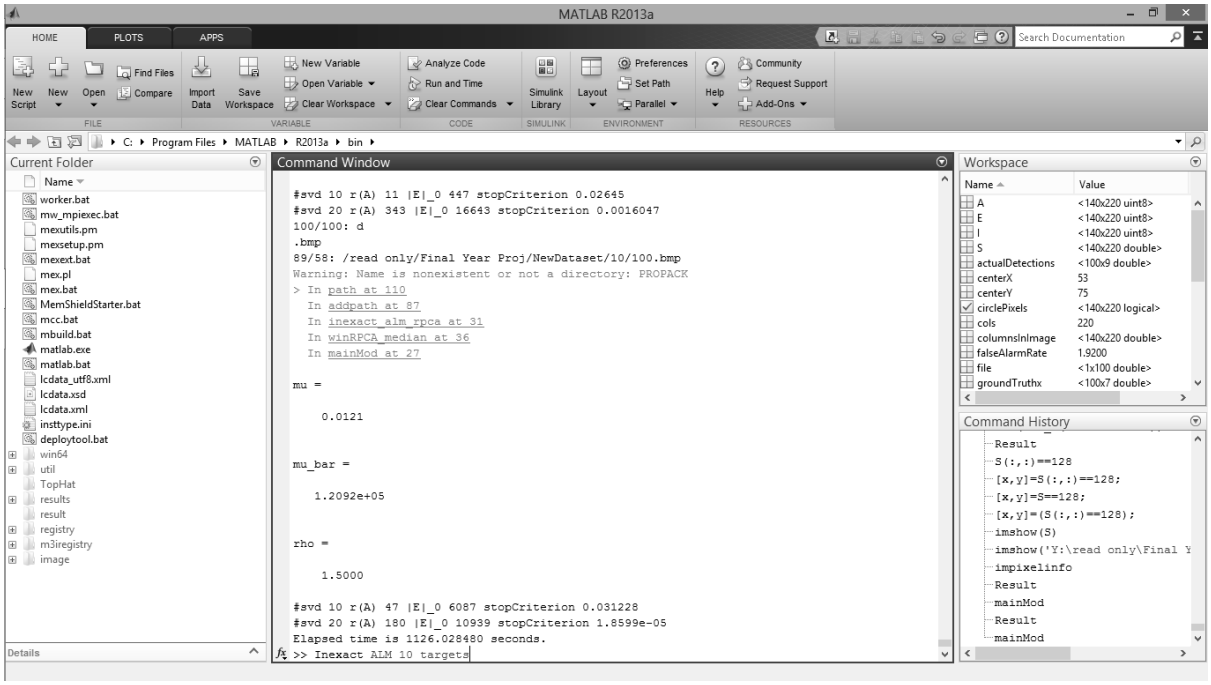
MATLAB Terminal Output (performance and time) after running Inexact ALM on 100 images with 4 targets





MATLAB Terminal Output (performance and time) after running Inexact ALM on 100 images with 7 targets





MATLAB Terminal Output (performance and time) after running Inexact ALM on 100 images with 10 targets

APPENDIX B: ACCOMPANYING CD

CD containing application software and source code has been distributed with this report. On it you will find the source code, and an electronic version of this report and power point presentation.

The source code is distributed as an '.m' file and requires that MATLAB 2013a or greater be installed. The CD contains a folder for each of the algorithm, segmentation, post processing, bipolar interpolation and target embedding. Each of these contains codes in MATLAB format

REFERENCES

[1] Chenqiang Gao, Deyu Meng, Yi Yang, et al., *"Infrared Patch-Image Model for Small Target Detection in a Single Image,"* Image Processing, IEEE Transactions on, vol.22, no. 12, pp. 4996-5009, 2013.

[2] T.-W. Bae, "*Small target detection using bilateral filter and temporal cross product in infrared images*," Infr. Phys. Technol., vol. 54, pp. 403–411, Sep. 2011.

[3] H. Deng, J. G. Liu, and Z. Chen, "*Infrared small target detection based on modified local entropy and EMD*," Chin. Opt. Lett., vol. 8, pp. 24–28, Jan. 2010.

[4] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Y. Ma, "*Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix*," Proc. CAMSAP, 2009, pp. 1–18.

[5] Y. Cao, R. Liu, and J. Yang, *"Small target detection using two dimensional least mean square (TDLMS) filter based on neighbourhood analysis,"* Int. J. Infr. Millimeter Waves, vol. 29, no. 2, pp. 188–200, 2008.

[6] X. Bai, F. Zhou, Y. Xie, and T. Jin, *"Modified top-hat transformation based on contour structuring element to detect infrared small target,"* in Proc. 3rd IEEE Conf. ICIEA, Jun. 2008, pp. 575–579.

[7] D. Liu, J. Zhang, and W. Dong, "*Temporal profile based small moving target detection algorithm in infrared image sequences*," Int. J. Infr. Millimeter Waves, vol. 28, no. 5, pp. 373–381, 2007.

[8] B. Zhang, T. Zhang, Z. Cao, and K. Zhang, "*Fast new small-target detection algorithm based on a modified partial differential equation in infrared clutter*," Opt. Eng., vol. 46, no. 10, pp. 106401-1–106401-4, 2007.

[9] Mahulikar, S.P., Sonawane, H.R., & Rao, G.A.: (2007) *"Infrared signature studies of aerospace vehicles"*, Progress in Aerospace Sciences, v. 43(7-8), pp. 218-245

[10] S. Deshpande, M. Er, V. Ronda, and P. Chan, "*Max-mean and Max-median filters for detection of small-targets*," Proc. SPIE, vol. 3809, pp. 74–83, Oct. 1999.

[11] P. A. Ffrench, J. R. Zeidler, and W. H. Ku, *"Enhanced detectability of small objects in correlated clutter using an improved 2D adaptive lattice algorithm,"* IEEE Trans. Image Process., vol. 6, no. 3, pp. 383–397, Mar. 1997.

[12] J. Silverman, J. M. Mooney, and C. E. Caefer, "*Tracking point targets in cloud clutter*," Proc. SPIE, vol. 3061, pp. 496–507, Aug. 1997.

[13] J. F. Rivest and R. Fortin, "*Detection of dim targets in digital infrared imagery by morphological image processing*," Opt. Eng., vol. 35, pp. 1886–1893, Jul. 1996.

[14] P. Wei, B. Zeidler, and W. Ku, "*Analysis of multiframe target detection using pixel statistics*," IEEE Trans. Aerosp. Electron. Syst., vol. 31, no. 1, pp. 238–247, Jan 1995.

[15] T. Soni, J. R. Zeidler, and W. H. Ku, "*Performance evaluation of 2D adaptive prediction filters for detection of small objects in image data*," IEEE Trans. Image Process., vol. 2, no. 3, pp. 327–340, Jul. 1993.

[16] V. Tom, T. Peli, M. Leung, and J. Bondaryk, *"Morphology-based algorithm for point target detection in infrared backgrounds,"* Proc. SPIE, vol. 1954, pp. 25–32, Oct. 1993.

[17] J. Ardouin, "*Point source detection based on point spread function symmetry*," Opt. Eng., vol. 32, pp. 2156–2164, Sep. 1993.

[18] B. Porat and B. Friedlander, "*A frequency domain algorithm for multiframe detection and estimation of dim targets*," IEEE Trans. Pattern Anal. Mach. Intell., vol. 12, no. 4, pp. 398–401, Apr. 1990.

[19] S. Reed, R. M. Gagliardi, and L. B. Stotts, "*Optical moving target detection with 3D matched filtering*," IEEE Trans. Aerosp. Electron. Syst., vol. 24, no. 4, pp. 327–336, Jan. 1988.

[20] https://en.wikipedia.org/wiki/Infra-red_search_and_track