

ANDROID BASED SHIP TRACKING SYSTEM

SARANYA.S

10PCA20

**A Project Report submitted to Avinashilingam Deemed
University for Women, Coimbatore in partial fulfillment of the
requirements for the Master's Degree in Computer Applications**

MAY-2013

**SIGNATURE OF THE
HEAD OF THE DEPARTMENT**

**SIGNATURE OF THE
SUPERVISOR**

**SIGNATURE OF THE
EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

I would like to express my sincere thanks to God Almighty, for his constant love and grace that he has showered upon me.

I am very grateful to **Dr.T.S.K.Meenakshi Sundaram, M.A., M.Phil., Ph.D.,** Chancellor, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for his support and encouragement during the course of my study.

I heartily thank **Dr.(Mrs.)Sheela Ramachandran M.Sc., P.G.Dip., Ph.D.,** Vice Chancellor Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for extending all resources that facilitated the conduct of the present study.

I express my humble gratitude to **Dr.(Mrs.) Gowri Ramakrishnan M.Sc., M.Phil., Ph.D.,** Registrar Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for providing all facilities necessary for the study.

I am also thankful to **Dr.(Mrs.) R. Parvatham M.Sc., Dip.Ed. M.Phil., Ph.D.,** Dean Faculty of Science, for granting the facility required.

I wish to place on record my deep sense of gratitude to **Dr.(Mrs.)G.Padmavathi M.Sc., M.Phil., Ph.D.,** Professor and Head, Department of Computer Science, for providing all the facilities to complete the project.

I take this unique opportunity to express my sincere thanks to my project Coordinator **Dr.(Mrs.)V.Srividhya, M.Sc., M.Phil., Ph.D., Assistant Professor, Department of Computer Science,** for her kind advice and knowledgeable suggestion, which helped me to complete my project successfully.

I owe great deal of gratitude to my esteemed guide **Dr. (Mrs.) P.Subashini, M.C.A., M.Phil., Ph.D., Professor,** Department of Computer Science, for her guidance, imparting the tremendous assistance and knowledgeable suggestion, which helped me to complete my project successfully.

I am greatly indebted to respected madam **Dr.Ms.M.Krishnaveni, M.Sc., M.Phil., Ph.D.**, Department of Computer Science Research Assistant for her inspiring and constant encouragement and timely guidelines at every stage of this project and who helped me to complete this project a successful one.

I have great pleasure in expressing my deep sense of gratitude to all other staffs and non-teaching staffs who stood behind the screen in making of project.

I would extend my hearty thanks to one and all that helped me directly or indirectly for successful completion of my project.

Last yet importantly, I would like to thank my parents, my brother and all my well-wishers for their kind inspiration.

CONTENTS

	Page No
1. INTRODUCTION	1
1.1 ABOUT THE SYSTEM	1
1.2 OVERVIEW OF THE PROJECT	2
2. SYSTEM CONFIGURATION	3
2.1 HARDWARE SPECIFICATION	3
2.2 SOFTWARE SPECIFICATION	3
2.3 ABOUT THE SOFTWARE	4
3. SYSTEM STUDY AND ANALYSIS	8
3.1 EXISTING SYSTEM	8
3.2 PROPOSED SYSTEM	8
4. SYSTEM DESIGN	9
4.1 DATASET USED	9
4.2 INPUT DESIGN	9
4.3 OUTPUT DESIGN	10
5. SYSTEM DEVELOPMENT	11
5.1 LOADING THE VIDEO	12
5.2 PREPROCESSING	12
5.3 SEGMENTATION	15
5.4 KALMAN FILTER BASED TRACKING	16

6. RESULTS AND DISCUSSION	20
7. CONCLUSION	21
8. SCOPE FOR FUTURE ENHANCEMENT	22
BIBLIOGRAPHY	23
APPENDIX	
9.1 SYSTEM FLOW DIAGRAM	
9.2 DATASET USED	
9.3 INPUT SCREENS	
9.4 OUTPUT SCREENS	
9.5 RESULTS	
9.5.1 LOADING THE VIDEO	
9.5.2 PREPROCESSING	
9.5.3 SEGMENTATION	
9.5.4 KALMAN FILTER BASED TRACKING	

SYNOPSIS

Android based ship tracking system helps to track the ship using Android software. Android is a Linux-based operating system. It is based on direct manipulation. As android comes with a vast library of useful functions, including functions for user interfaces, image/bitmap manipulations it proves to be an efficient platform for image processing. Android based image processing paves way for various applications. Image processing is a form of signal processing for which the input can be an image, such as a photograph or video frame; the output may be either an image or a set of characteristics or parameters related to the image. Inorder to perform image processing in android platform, a special tool named matclipse is used. Object tracking is the problem of estimating the positions and other relevant information of moving objects in image sequences. Object tracking includes many obvious applications like perception and control for autonomous surveillance systems, identifying and neutralizing threats in missile defense, optimizing traffic control systems, and improving human-computer interaction.

This project focuses on using Android platform to track a ship in a SAR video using a tool named matclipse. The tracking is performed using a filter named Kalman filter. It has its own characteristics to track the ship. The main purpose of using Android platform is to apply it on real time basis i.e; it can do real-time tracing the location of target objects. In order to track the ship, the video is first preprocessed using contrast enhancement and noise removal techniques and then segmented using background subtraction model. Kalman filters can be applied to many different types of systems and can refer to any measurable quantity, such as an object's location, velocity, temperature, voltage, or a combination of these.

1. INTRODUCTION

1.1 ABOUT THE SYSTEM

This section is used to describe the problem definition, overview of the project and the purpose for which the project is developed and used.

The main goal of Android based ship tracking system is to track a ship based on certain geometric features such as centre and area of the ship using matclipse tool in android software, so that it can be used for real time purpose. The tracking of ship is performed using Kalman filter, one of the most popular filter used for tracking objects.

Tracking of ship in SAR image is performed using Image processing techniques with Matclipse as software tool. Matclipse is an open source project which is an interface between the Java integrated development environment (IDE) Eclipse and Matlab.

The aim is to develop a system for tracking ship from SAR image. It is performed by using some of the image processing techniques such as preprocessing, segmentation and tracking.

MAIN OBJECTIVES

The objectives of ship tracking is,

- To develop a ship tracking system in android platform.
- To develop an Image processing based applications in android platform.
- To track a ship based on certain features such as centre and area of the ship using Kalman filter.

1.2 OVERVIEW OF THE PROJECT

Tracking of ship in android platform is processed using some of the image processing techniques in matclipse software. In order to track a ship the following steps are carried out

- Preprocessing
- Segmentation
- Tracking

Preprocessing is one of the major step to be performed on the video before the object to be tracked is segmented. In this project, filtering (Contrast Enhancement) and noise removal are the two major preprocessing techniques performed. Presence of noise in video degrades its quality. In order to recover from this problem the preprocessing techniques are performed.

The main goal is to track a particular object in the video. Thus the initial step is to segment the desired object using certain segmentation techniques. In this project the object is segmented based on background subtraction process.

Finally the segmented object is to be tracked. Kalman filter based tracking is performed in this project. This is one of the efficient filter used to track a particular object.

2. SYSTEM CONFIGURATION

This section describes the hardware and software specification needed for both development and implementation phases of this project.

2.1 HARDWARE SPECIFICATION

Processor	: Intel® Core(TM)i3-2310M CPU @ 2.10GHz 2.10 GHz
RAM	: 4.00GB
System type	: 32 –bit Operating System
Keyboard	: Standard PS/2 key board

2.2 SOFTWARE SPECIFICATION

Software	: Android SDK, Java JDK, Eclipse IDE,Matclipse
Operating System	: Microsoft Windows 7

2.3 ABOUT THE SOFTWARE

Android SDK

The Android SDK provides you the API libraries and developer tools necessary to build, test, and debug apps for Android. In order to develop an Android application for the first time it is recommended to download the ADT Bundle .It includes the essential Android SDK components and a version of the Eclipse IDE with built-in ADT (Android Developer Tools)

The ADT Bundle includes

- Eclipse + ADT plugin
- Android SDK Tools
- Android Platform-tools
- The latest Android platform
- The latest Android system image for the emulator

Java JDK

The Java SE Development Kit (JDK) is another essential tool.It provides all the tools needed to develop and run Java programs. The executable (.exe file) is used for installation.

Eclipse IDE

Android offers a custom plugin for the Eclipse IDE, called Android Development Tools (ADT). This plugin provides a powerful, integrated environment in which to develop Android apps.This makes the platform for building the Android projects easily.

Matclipse

Matclipse is an interface between the Java integrated development environment (IDE) Eclipse and Matlab. The aim of the project is to facilitate the usage of Matlab directly from Eclipse under various operating systems (Linux, Windows, OS X). At the moment, it consists of a workbench, a Matlab editor, a Matlab console including a command window and a viewer for results, a Matlab command history and a Matlab workspace viewer.

Matlab Workbench

The Matlab workbench uses a tree structure for workspace. One can create Matlab projects there and, of course, all basic features from Eclipse are supported there. Projects can have a Matlab nature and files can be executed directly from the workbench without opening them in the editor.

Communication with Matlab

The communication between Java and Matlab is based on the usage of pipes under Linux and OS X and on the usage of DCOM under Windows. In both cases it is necessary to have a local installation of Matlab.

Matlab Editor

The Matlab editor is based on a regular expression parser which provides syntax highlighting and checking.

The current features are:

- Smart indentation and parenthesis matching.
- Detection of variables, operators, strings, commands and key words.
- Integration of the Matlab program mlint to mark errors and warnings directly in the editor.

Possible improvements for the future could be:

- Syntax highlighting with detection of all Matlab commands also from available toolboxes. The list of these commands has to be extracted from the Matlab help directory for a given version of Matlab. Commands from Matlab directly and from toolboxes can be shown in different colors.
- Direct link to Matlab help pages locally or situated at MathWorks.
- Evaluation of the whole file or of a selection of lines directly in Matlab.
- Export of a documented file in XML, Latex, PDF or HTML format. With syntax highlighting, indentation and links to Matlab help.
- Usage of a simple wiki-like markup language to display mathematical formulas, lists

Matlab Console

In contrast to Matlab where input to and output from Matlab is displayed in the same window, *matclipse* uses a concept with a separate command line for input to Matlab and a results viewer for displaying the output from Matlab nicely separated by the input commands.

The current features of the command line are the following:

- Syntax highlighting is provided by the editor.
- Up- and down-scrolling in the command history.
- The Matlab command pause without an additional time specified had to be overwritten. It is terminated now automatically after 2 seconds because there was no way to terminate the pause from the command line. In future one would have to think of an icon (or a shortcut) which would terminate the pause in similar way as CTRL-C is handled to stop execution of a command.
- An icon was provided in the console to stop execution of Matlab.
- Icons are provided to start and stop Matlab.
- Additional icons are available to bring figures in the foreground or to delete figures.
- There is a possibility to change the Matlab directory without a Matlab command.

- One can start the Matlab help browser from an icon.
- One can clear the results view from an icon (*clc* does not work).
- One can toggle Matlab console debug from an icon, to debug the XML communication between *matclipse* and Matlab. Matlab Command History.

The Matlab history records commands which are executed for further usage. The data are stored in sections with different starting times of a working session. In contrast to Matlab, lines which are used again without any change are not duplicated but moved to the current location. This keeps the list shorter.

In case a couple of lines are executed from the editor with `execute selection`, those lines are stored in one entry in the command history and can be recalled at once.

Matlab Workspace Viewer

The Matlab workspace viewer enables the user to explore the Matlab variables. In contrast to Matlab, a tree is used for this view where all data types can be displayed. So called container data types like cells and structures are displayed with their size and then a sub-tree is opened which shows the contents in the container. This is possible for the whole depths of those data types. To save time especially when working with pipes, first only the structure of the workspace is shown in the tree together with size information. The detailed information about the content of an array is only transferred from Matlab to Eclipse when this entry is chosen with the mouse or the keyboard.

The exchange of information between Matlab and Eclipse is based on a XML description of the workspace generated within Matlab.

3. SYSTEM STUDY AND ANALYSIS

System development can generally be thought of having two major components: system analysis and systems design. In system Analysis more emphasis is given to understanding the details of an existing system or a proposed one and then deciding whether the proposed system is desired or not and whether the existing system needs improvements. Thus, system analysis is the process of investigating a system, identifying problems, and using the information to recommend improvements to the system.

3.1 Existing System

The existing system for tracking process is performed in MATLAB. In this project it is performed using Eclipse environment with the help of matclipse tool so that it can be used for real time processing.

3.2 Proposed System

In the proposed system, an object in a video is tracked on real time basis i.e., a particular video is loaded in the emulator and preprocessed to enhance the quality of the video and then the object to be tracked is segmented and finally tracked using kalman filter.

Merits of proposed system

- Android platform enables to use the project for real time purpose.
- Implementation of Image Processing using Android platform.
- Used for video surveillance.

4. SYSTEM DESIGN

Design is the first process in the development phase of any engineered system. The inputs to the design are the software requirements and the output will be the design specification applicable to all software.

4.1 DATASET USED

In this project, simulated SAR ship images are converted into video. This video along with the extracted features of the object in the video which is to be tracked are considered as the dataset since the project deals with tracking. Android based ship tracking system is capable of tracking the SAR ship target according to the extracted features on real time basis. In this system the video is taken as input and each of the video frames are considered for preprocessing which is shown in APPENDIX [9.2.1] and [9.2.2]. Then the preprocessed video frames are taken for segmentation of the object to be tracked which is shown in APPENDIX [9.2.3]. Finally the frames with segmented object is taken for tracking which is shown in APPENDIX [9.2.4].

4.2 INPUT DESIGN

The main objective of the input design is to provide user friendly interaction. As the main goal is to track an object present in a video on real time basis, the dataset considered is the video, where the video frames are taken as input along with the extracted features. Input form is designed by using Eclipse IDE GUI and is shown in APPENDIX [9.3]. These video frames are considered as input for preprocessing steps. For the first step, to enhance the contrast for each video frames individually each frames are taken which is shown in APPENDIX [9.2.1]. In order to denoise, the contrast enhanced frames are taken which is shown in APPENDIX [9.2.2]. In order to segment the moving object, the preprocessed frames are taken which is shown in APPENDIX [9.2.3]. Finally to track, the frames with segmented object is taken and is shown in APPENDIX [9.2.4]. All these dataset are fed into the system using matclipse.

4.3 OUTPUT DESIGN

Output design generally refers to the results and information that are generated by the system.

The video is loaded and the video frames are considered as input for further processing. Various techniques of image processing are applied to generate variety of output. In this project, the final output is the video in which the object is been tracked using kalman filter which is shown in APPENDIX [9.5.4].

The video frames are fed into the preprocessing stage in which the contrast is enhanced and the noise is filtered using specific methods. The output of the preprocessing step is shown in APPENDIX [9.5.1 and 9.5.2].

The object in the video is segmented using the background subtraction model and the geometric features of the object is used to track the object frame by frame. The output of the segmentation process is shown in APPENDIX [9.5.3].

Finally the tracking of the object is performed and the output is shown in APPENDIX [9.5.4].

5. SYSTEM DEVELOPMENT

System development is a series of operations performed to manipulate data to produce output from a computer system. The principle activities performed during the development phase can be divided into a major related sequence. They are

- Internal
- External

The major system development activities done for the system is computer program development and performance testing.

The major external system development activities done are planning and implementation.

Module Description

The project consist of five main modules namely

1. Loading the video
2. Preprocessing
3. Segmentation
4. Kalman filter based tracking

Object tracking is a process of locating the object to associate the target in successive video frame over time and it finds wide scale applications in the field of security and surveillance, video communication, augmented reality, traffic control, medical imaging etc. The aim of motion tracking is to detect and track moving objects through a sequence of images.

5.1 LOADING THE VIDEO

The most important part of the project is played by the video which has the object to be tracked. The first module is to load the video for further processing. In order to perform this process the required video has to be loaded and its frames are considered individually for each process so that during execution the video can be displayed in the screen. The video loaded is shown in APPENDIX [9.4.1].

5.2 PREPROCESSING

Videos may suffer from the following degradations:

- Poor contrast due to poor illumination or finite sensitivity of the device
- Electronic sensor noise or atmospheric disturbances leading to broad band noise
- Aliasing effects due to inadequate sampling
- Finite aperture effects or motion leading to spatial

Preprocessing videos commonly involves removing low-frequency background noise, normalizing the intensity of the individual particles of videos, removing reflections, and masking portions of videos. Video preprocessing is the technique of enhancing data videos prior to computational processing. It is a method to enhance some of the features of the videos for further processing. It does not increase the information content but increases the quality of the video. A number of preprocessing techniques are available. Preprocessing is performed on each of the frames of the loaded video.

The preprocessing techniques used in this project are

- Filtering (Contrast Enhancement)
- Noise removal

5.2.1 FILTERING (CONTRAST ENHANCEMENT)

Video enhancement is one of the most important and difficult components in video research. The aim of video enhancement is to improve the visual appearance of the video, or to provide a “better” transform representation for future automated video processing, such as analysis, detection, segmentation, recognition, surveillance, traffic. When a camera is used to capture a video, the resultant output is a blurred one. Carrying out video enhancement understanding under low quality video is a challenging problem.

Video enhancement techniques involve processing an image/frame to make it look better to human viewers. It is usually used for post processing by modifying contrast or dynamic range or both in an image. The aim of contrast enhancement process is to adjust the local contrast in different regions of the image so that the details in dark or bright regions are brought out and revealed to the human viewers. Contrast enhancement is usually applied to input images to obtain a superior visual representation of the image by transforming original pixel values.

Filtering is a technique for modifying or enhancing. When a contrast enhancement technique is applied to a video it improves the perceptibility of objects in the scene by enhancing the brightness difference between objects and their backgrounds. Enhancement technique can be divided into two categories namely,

- Spatial domain
- Frequency domain

In this project spatial domain technique is applied to enhance the intensity values in a particular video. It increases the contrast of the image by mapping the values of the input intensity image to new values such that, by default, 1% of the data is saturated at low and high intensities of the input data. The loaded video is taken as input for contrast enhancement based on spatial domain. Each of the frames is enhanced individually so that the intensity values is enhanced and thus the video is completely enhanced. The output of the contrast enhanced video is shown in APPENDIX [9.4.2].

5.2.2 NOISE REMOVAL

Reducing noise from the satellite image is a challenge for the researchers in digital image processing. Several approaches are there for noise reduction. Noise reduction is the process of removing noise from a signal. All recording devices, both analogue or digital, have traits which make them susceptible to noise. The presence of noise in videos degrades its quality and makes it complicated for further processing. Video denoising is the process of removing noise from a video signal. This is performed by applying reduction methods frame by frame individually.

While a video is captured, it is very much exposed to noise and is easily disturbed with noise signals. Different types of noise are present. In this project, speckle noise is handled, that is present in the video. Speckle noise is a granular noise that inherently exists in and degrades the quality of the video. It results from random fluctuations in the return signal from an object that is no bigger than a single image-processing element. It increases the mean grey level of a local area. Speckle noise in SAR is generally more serious, causing difficulties for interpretation. It is caused by coherent processing of backscattered signals from multiple distributed targets. Speckle noise is commonly found in synthetic aperture radar images, satellite images etc.,.

Different types of filtering approaches are present in order to remove noise. In this project, median filter is used to remove the speckle noise present in the video. The median filter is a nonlinear digital filtering technique, often used to remove noise. Such noise reduction is a typical pre-processing step to improve the results of later processing. Median filtering is very widely used in digital image processing because, under certain conditions, it preserves edges while removing noise. The main idea of the median filter is to run through the signal entry by entry, replacing each entry with the median of neighboring entries. The pattern of neighbors is called the "window", which slides, entry by entry, over the entire signal. Median filtering is one kind of smoothing technique. Each of the individual frames which has been undertaken the contrast enhancement technique is taken as input, to these frames median filter is used. Each of the frames present in the video is filtered (smoothed) using median filter and the noise is removed. The filter is

applied to each of the frames individually. The output screen of the noise removed video are shown in APPENDIX [9.4.3] and the results are shown in APPENDIX [9.5.2].

5.3 SEGMENTATION

Key purpose of video segmentation is to enable content-based representation by extracting objects of interest from a series of consecutive video frames. Mainly, it is required for high-level image understanding and scene explanation such as spotting and tracking of special events in surveillance video. Tracking the object in a video can be done easily with the help of the segmentation methods. Some of the methods available for object segmentation are background subtraction, temporal segmentation, edge detection, spatial segmentation and optical flow.

Background subtraction is a commonly used method for segmenting out objects of interest in a scene. The name “background subtraction” comes from the simple technique of subtracting the observed image from the estimated image and threshold the result to generate the objects of interest.

The approach is to perform background subtraction, which identifies moving objects from the difference between the current frame and a reference frame.

5.3.1 BACKGROUND SUBTRACTION MODEL

Background subtraction is a commonly used class of techniques for segmenting out foreground objects from the background in a sequence of video frames. It segments out objects of interest in a scene for applications such as surveillance. This method is useful for detecting the moving objects in a surveillance camera. By using this method the object can be tracked perfectly. The core of each motion detection system is the part of background subtraction that effectively extracts the correct shape of moving objects. It involves comparing an observed image with an estimate of the image if it contained no objects of interest. The areas of the image plane where there is a significant difference between the observed and estimated images indicate the location of the objects of interest.

The name “background subtraction” comes from the simple technique of subtracting the observed image from the estimated image and thresholding the result to generate the

objects of interest. Background subtraction is a widely used approach for detecting moving objects in videos. Background subtraction is mostly done if the image is a part of a video stream. Different types of approaches are present for background subtraction based segmentation. In this project, frame differencing method is used which mainly depends upon the time intervals. The background is assumed to be the frame at time t and the frame difference at time $t + 1$ is calculated using the function. This difference image would only show some intensity for the pixel locations which have changed in the two frames. A threshold "Th" is put on this difference image to improve the subtraction. The accuracy of this approach is dependent on speed of movement in the scene. Faster movements may require higher thresholds. The output screen of the segmented object is shown in APPENDIX [9.4.4]. The use of background subtraction method is to extract the foreground target as the initialization of the tracking algorithm.

5.4 KALMAN FILTER BASED TRACKING

Tracking is the problem of generating an inference about the motion of an object given a sequence of images. The two steps in video analysis are

1. Detection of interesting moving objects through segmentation.
2. Tracking of such objects from frame to frame

There is a strong relationship between the object representations and the tracking algorithms. Object representations are usually chosen according to the application domain.

Objects can be represented by their

- Shapes and appearances
- Geometric measures

In a long image sequence, if the dynamics of the moving object is known, prediction can be made about the positions of the objects in the current image. This information can be combined with the actual image observation to achieve more robust results.

In order to perform the tracking process certain features has to be extracted. In this project the objects' (to be tracked) centre and area are found out.

CENTRE

The centre of an object is a point in the middle of the object. It is a fixed point of the isometrics.

AREA

Area is amount of surface the 2D shapes cover. It is measured in square unit. A mathematically acceptable definition of area is complex. Area is usually measured or defined on a flat surface, also called a Euclidean plane, or on a spherical surface. The surface area is occasionally determined for irregular objects. In the case of extremely complex or esoteric surfaces, the area might be impossible to define or measure.

The Kalman filter is a framework for predicting a process's state, and using measurements to correct or 'update' these predictions. It is an optimal estimator - ie infers parameters of interest from indirect, inaccurate and uncertain observations. It is recursive so that new measurements can be processed as they arrive. The solution is recursive in that each updated estimate of the state is computed from the previous estimate and the new input data, so only the previous estimate requires storage.

WHY IS KALMAN FILTERING SO POPULAR?

- ✓ Good results in practice due to optimality and structure.
- ✓ Convenient form for online real time processing.
- ✓ Easy to formulate and implement given a basic understanding.
- ✓ Measurement equations need not be inverted.

5.4.1 OBJECT TRACKING SYSTEM

The goal of an object tracking is to generate the trajectory of an object over time by discovering its exact position in every frame of the video sequence. The algorithm for object tracking is composed of two steps: selection object module in the first frame of

the sequence and then applying kalman filter. The selection module selects the position of the object in the first frame. It consists of extracting the module initialization parameters that are moving through the position, size, width, length of the search window of the object in the first frame of the sequence.

5.4.2 KALMAN FILTER IMPLEMENTATION

The Kalman filter estimates the position of the object in each frame of the sequence. The input parameters of the Kalman filter, respectively, the position of the object in the image at time k , the size of the object and the width and length of the search window of the object which vary due to the mobility of the object during the sequence. These parameters represent the state vector and measurement vector of the Kalman filter. In general, the estimation of parameters followed with a Kalman filter is a process that requires the following steps:

- The measure is to take the tracking parameters.
- The estimate, which updates the position of the object.
- The prediction, which calculates the position of the object in the next frame.

In target tracking applications, the most popular methods for estimating target positions is the Kalman filter. The Kalman filter assumes that the dynamics of the target can be modeled and that the noise affecting the target dynamics and the sensor data is stationary with zero mean. The Kalman filter is a recursive estimator. This means that only the estimated states from the previous time step and the current measurements are needed to estimate the current states.

The major steps performed are,

- Initialization for Kalman Filtering.
- Initialization of the Kalman filter parameters.
- Looping over all image frames in the video.

To use Kalman filter for object tracking we assume that the motion of the object is almost constant over frames. The Kalman filter is a recursive predictive filter that is based on the use of state space techniques and recursive algorithms. It estimates the state of a dynamic system. This dynamic system can be disturbed by some noise, mostly assumed as white

noise. To improve the estimated state the Kalman filter uses measurements that are related to the state but disturbed as well. Thus the Kalman filter consists of two steps:

1. The prediction
2. The correction

In the first step the state is predicted with the dynamic model. In the second step it is corrected with the observation model, so that the error covariance of the estimator is minimized. In this sense it is an optimal estimator.

This procedure is repeated for each time step with the state of the previous time step as initial value. Therefore the Kalman filter is called a recursive filter.

Using these models the vectors are initialized for Kalman filter and then the centre and area is calculated for each frames. Using these measures the object is tracked along with the window size. The output screen and result for tracking is shown in APPENDIX [9.4.5] and [9.5.5].

6. RESULTS AND DISCUSSION

In this project the ship target is tracked on real time basis i.e., using Android eclipse and matclipse platform. The tracking of ship is performed using Kalman filter which is one of the efficient tracking filter. The result of tracking is shown in APPENDIX [9.5.5]. In order to track a particular object it has to undergo two important steps, Preprocessing and Segmentation. Preprocessing involves contrast enhancement and noise removal. The noise is reduced from the video using median filter. The results of preprocessing are shown in APPENDIX [9.5.1 and 9.5.2].

Segmentation is another important step which extracts the object of interest. In this project background subtraction model based segmentation is performed and the output is shown in APPENDIX [9.5.3]. All these steps are performed frame by frame continuously. The tracked object using Kalman filter is shown in APPENDIX [9.5.4]. The output screens for preprocessing, segmentation and tracking are shown in APPENDIX [9.4] individually.

7. CONCLUSION

The project “**Android based ship tracking system**“ is implemented successfully and it is more useful for video surveillance. The objective of this project is to track the ship target from a video using android eclipse platform and matclipse tool. It is more useful for marine traffic system and military surveillance for detecting and tracking the ship. The major advantage of this system is that it can be used for real time purpose. Also the preprocessing and segmentation techniques performed enhances the tracking procedure.

8. SCOPE FOR FUTURE ENHANCEMENT

Scope is a measure of considering a necessary part of a project. Scope is very important on large projects. But the real problem was often that too much effort was spent on unnecessary things. People especially volunteer coders and builders have a limited amount of free time and energy to work on projects.

Every application has its own merits and demerits. The project has covered almost all the requirements. Further requirements and improvements can easily be done since the coding is mainly structured or modular in nature. Changing the existing modules or adding new modules can append improvements. Further enhancement can be done to the applications so that the model functions very attractive and useful manner than the present one.

- The object can be tracked from a real time captured video.
- Other tracking filters can be implemented.
- Outputs can be displayed on the emulator.

9. BIBLIOGRAPHY

1. A. Alper and J. Omar and S. Mubarak, “ **Object Tracking: A Survey**” ACM Computing Surveys, vol. 38, no. 45, pp. 1–45. 2006.
2. Caius SULIMAN, Cristina CRUCERU and Florin MOLDOVEANU, ” **Kalman Filter Based Tracking in an Video Surveillance System**”, 10th International Conference on DEVELOPMENT AND APPLICATION SYSTEMS, Suceava, Romania, May 27-29, 2010.
3. M. Drew, J. Wei, and Z.-N. Li, “ **Illumination invariant image retrieval and video segmentation**,” Pattern Recog., vol. 32, no. 8, pp. 1369–1388, 1999.
4. Erik Cuevas, Daniel Zaldivar and Raul Rojas, ” **Kalman filter for vision tracking**”, August 2005.
5. G. Welch and G. Bishop, “ **An introduction to the Kalman filter**” Univ. North Carolina Chapel Hill, Tech. Rep., 1995.
6. W.L. Khong, W.Y. Kow, H.T. Tan, H.P. Yoong, K.T.K. Teo, ” **Kalman Filtering Based Object Tracking in Surveillance Video System**”, Proceedings of the 3rd (2011) CUTSE International Conference Miri, Sarawak, Malaysia, 8-9 Nov, 2011.
7. Milindkumar V. Sarode, Prashant R. Deshmukh, “ **Reduction of Speckle Noise and Image Enhancement of Images Using Filtering Technique**”, International Journal of Advancements in Technology <http://ijict.org/> ISSN 0976-4860, Vol 2, No 1 (January 2011) ©IJoAT.
8. P. Spagnolo, T.D’ Orazio, M. Leo, A. Distanto “ **Moving object segmentation by background subtraction and temporal analysis**”, Image and Vision Computing 24 (2006) pp. 411–423.
9. Ssu-Wei Chen, Luke K. Wang, Jen-Hong Lan, ” **Moving Object tracking Based on Background Subtraction Combined Temporal Difference**”, International Conference on Emerging Trends in Computer and Image Processing (ICETCIP'2011) Bangkok Dec.2011.

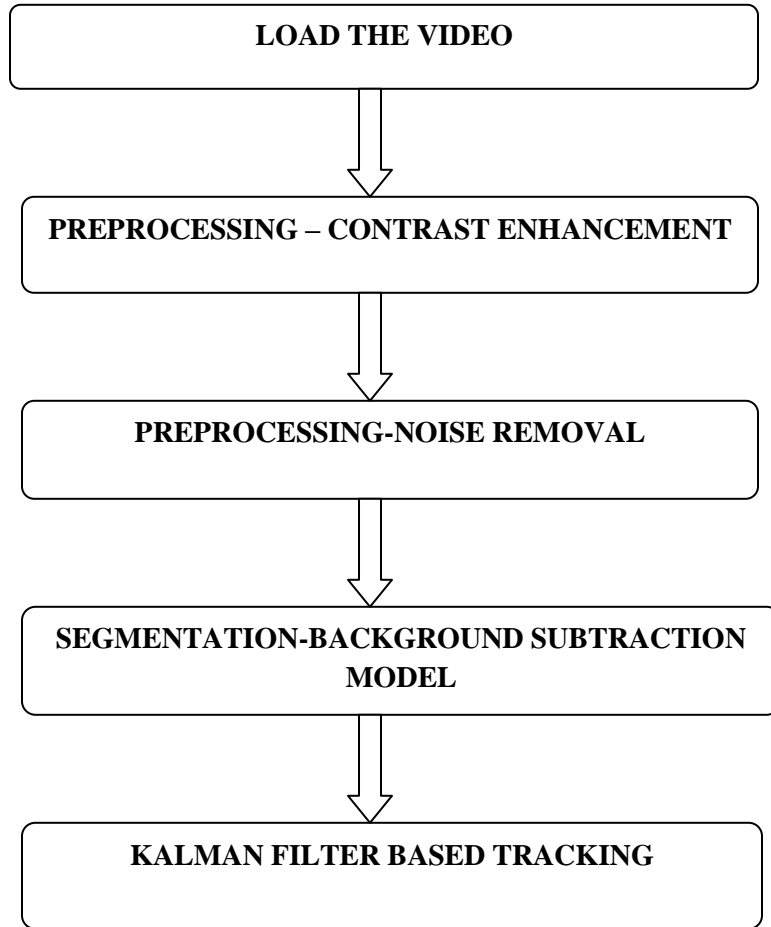
10. Yunbo Rao, Leiting Chen, “**A Survey of Video Enhancement Techniques**”,
Journal of Information Hiding and Multimedia Signal Processing Oc 2012 ISSN
2073-4212 ,Volume 3, Number 1, January 2012.
11. ZHIJUN QIU, DANYA YAO,” **Kalman Filtering Used in Video-Based Traffic
Monitoring System**”, Journal of Intelligent Transportation Systems ,10(1):15–21,
2006, Copyright C Taylor and Francis Group, LLC,ISSN: 1547-2450 print / 1547-
2442 online DOI: 10.1080/15472450500455211.

WEBSITES

1. http://www.cs.cornell.edu/Courses/cs4758/2011sp/final_projects/spring_2011/Xu_Chang.pdf
2. <http://cs229.stanford.edu/proj2007/Kim-ObjectTrackingInAVideoSequence.pdf>
3. http://www.eecis.udel.edu/~ryanz/BioControls/Home_files/CDC12_2221_FI.pdf
4. <http://blog.cordiner.net/2011/05/03/object-tracking-using-a-kalman-filter-matlab/>
5. <http://elib.uni-stuttgart.de/opus/volltexte/2005/2183/pdf/kleinbauer.pdf>
6. <http://classes.soe.ucsc.edu/cmpe264/Fall06/Lec15.pdf>
7. http://www.cs.cmu.edu/~motionplanning/papers/sbp_papers/integrated3/kleeman_kalman_basics.pdf
8. <http://www.waset.org/journals/waset/v64/v64-130.pdf>
9. <http://www.dasconference.ro/cd2010/data/papers/A47.pdf>
10. http://www.academia.edu/1767655/Kalman_Filtering_Based_Object_Tracking_in_Surveillance_Video_System
11. <http://www.tandfonline.com/doi/pdf/10.1080/15472450500455211>
12. <http://repositorio-aberto.up.pt/bitstream/10216/324/2/12767.pdf>
13. <http://www.stanford.edu/class/ee368/Android/Tutorial-1-Basic-Android-Setup-Linux-SCIEN.pdf>
14. <http://www.3programmers.com/mwells/documents/pdf/Final%20Report.pdf>

APPENDIX

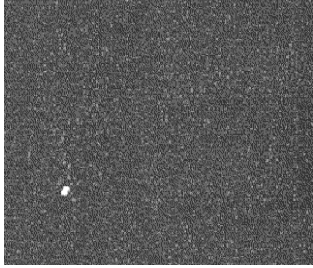

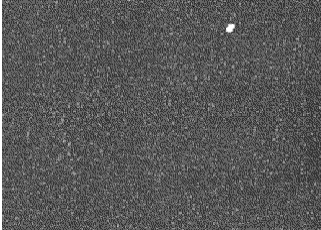
9.1 SYSTEM FLOW DIAGRAM

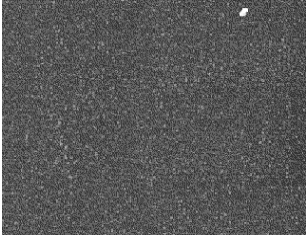
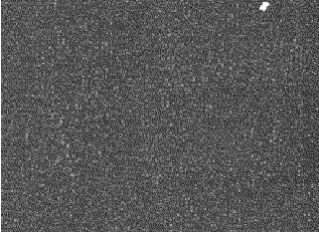


9.2 DATASET USED

9.2.1 DATASET USED FOR PREPROCESSING

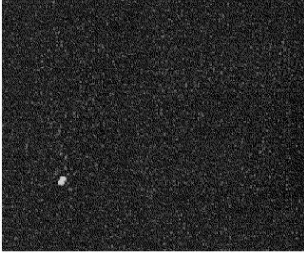
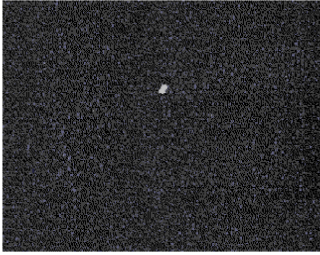
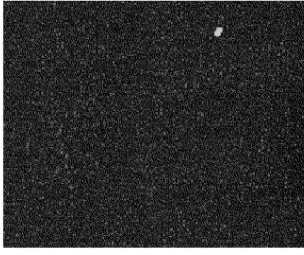
CONTRAST ENHANCEMENT

FRAME NUMBERS	INPUT VIDEO FRAMES
1	
2	
3	

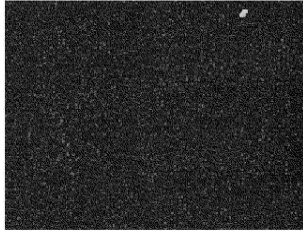
4	
5	

9.2.2 DATASET USED FOR PREPROCESSING

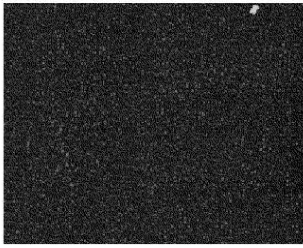
NOISE REMOVAL

FRAME NUMBERS	INPUT VIDEO FRAMES
1	
2	
3	

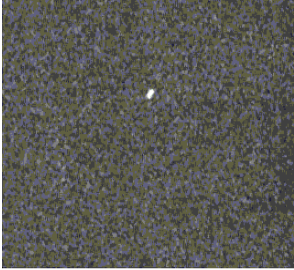
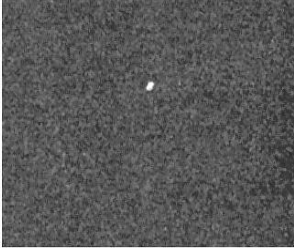
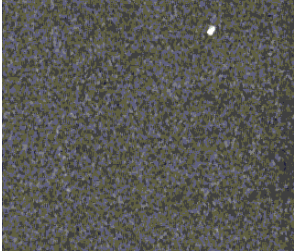
4



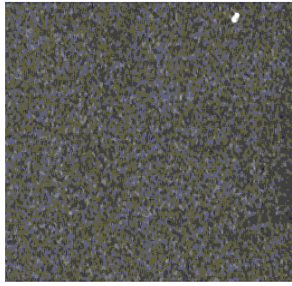
5



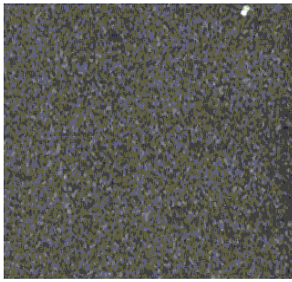
9.2.3 DATASET FOR SEGMENTATION

FRAME NUMBERS	INPUT VIDEO FRAMES
1	
2	
3	

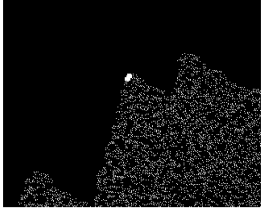
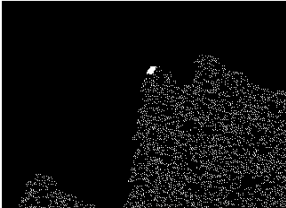
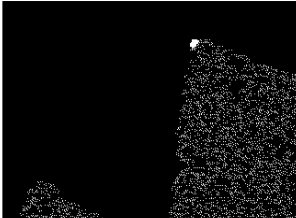
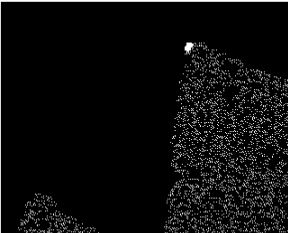
4



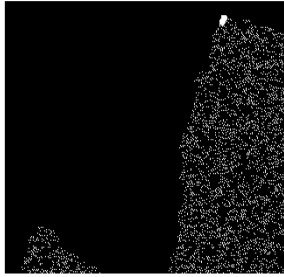
5



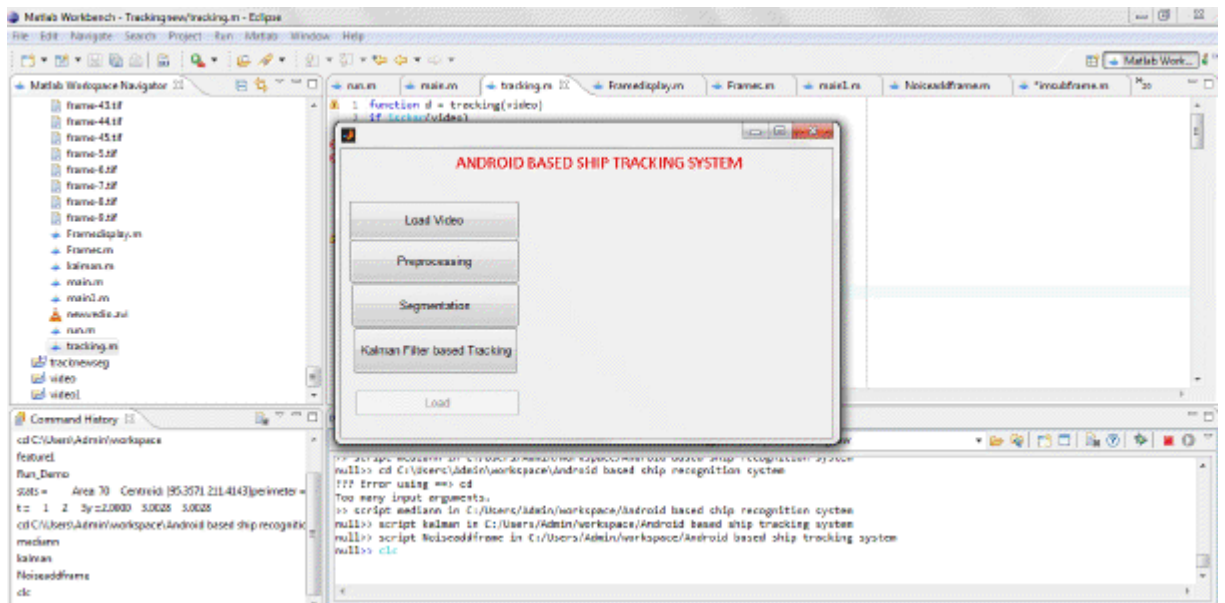
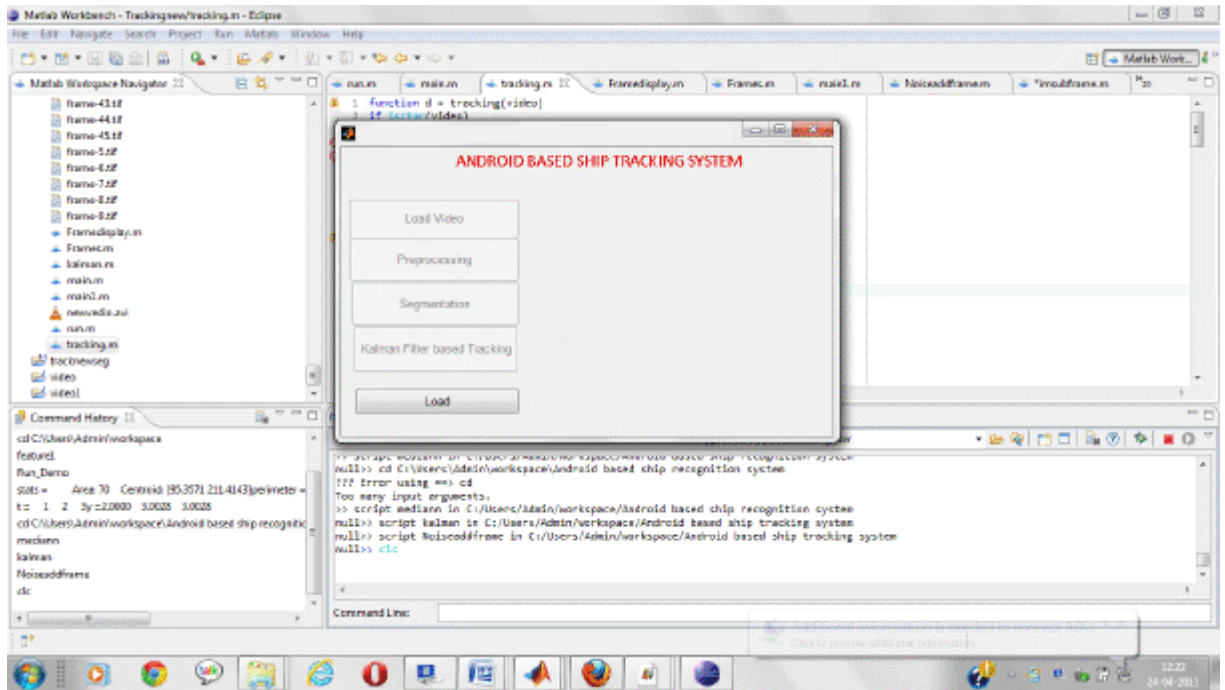
9.2.4 DATASET FOR TRACKING

FRAME NUMBERS	INPUT VIDEO FRAMES
1	
2	
3	
4	

5

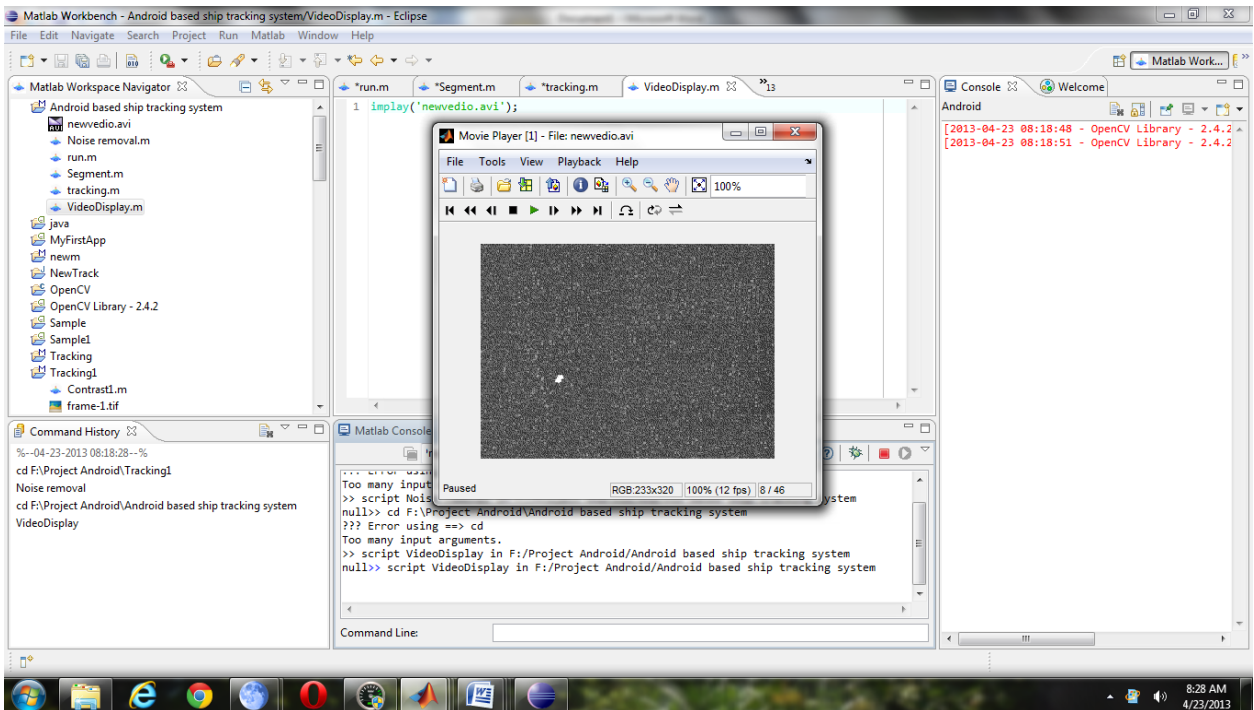
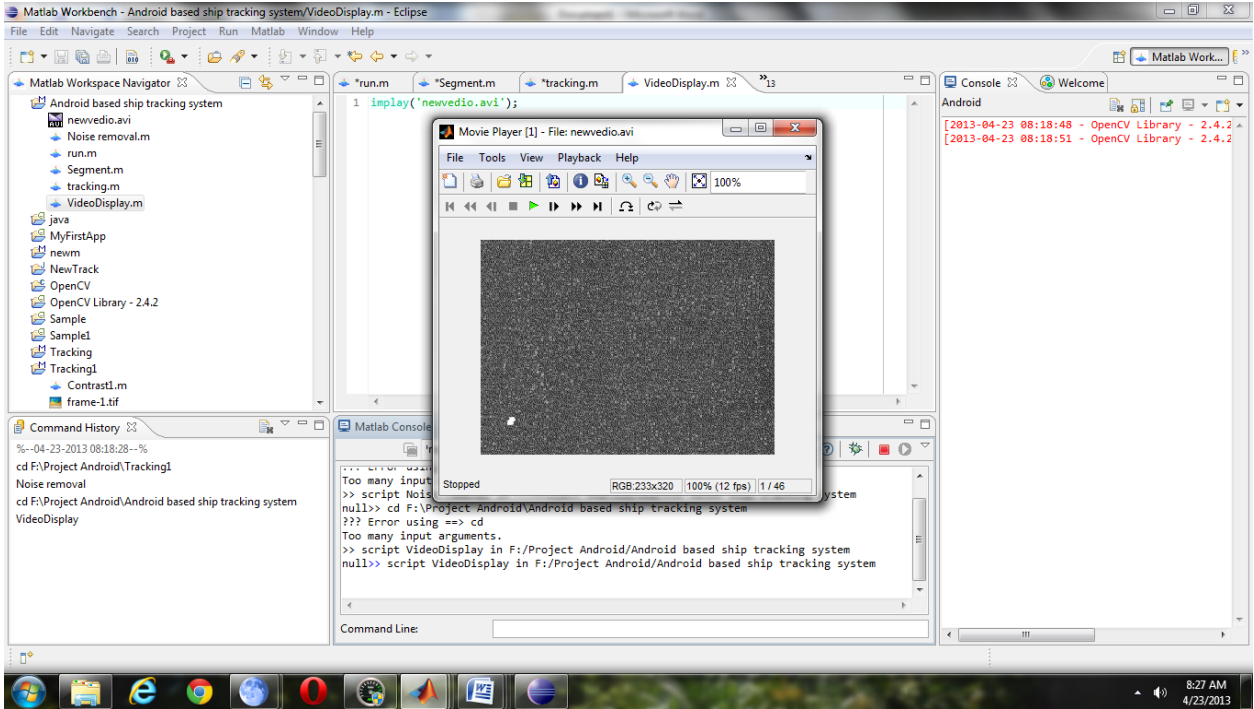


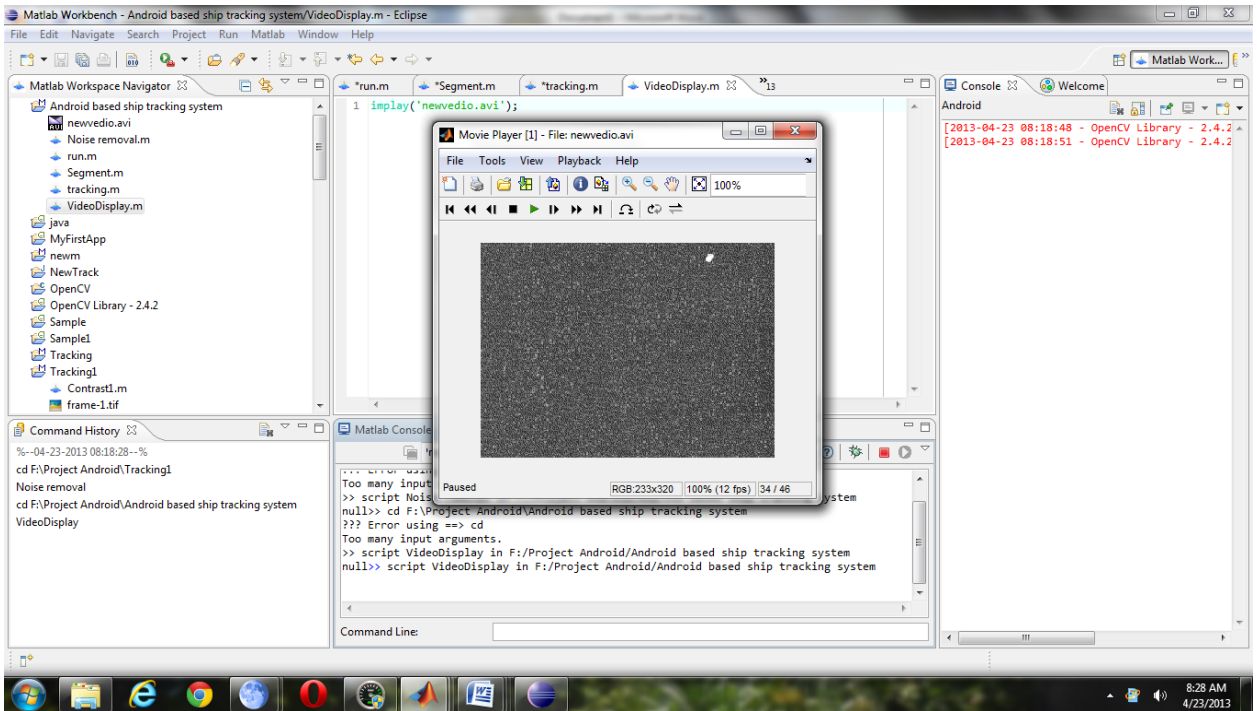
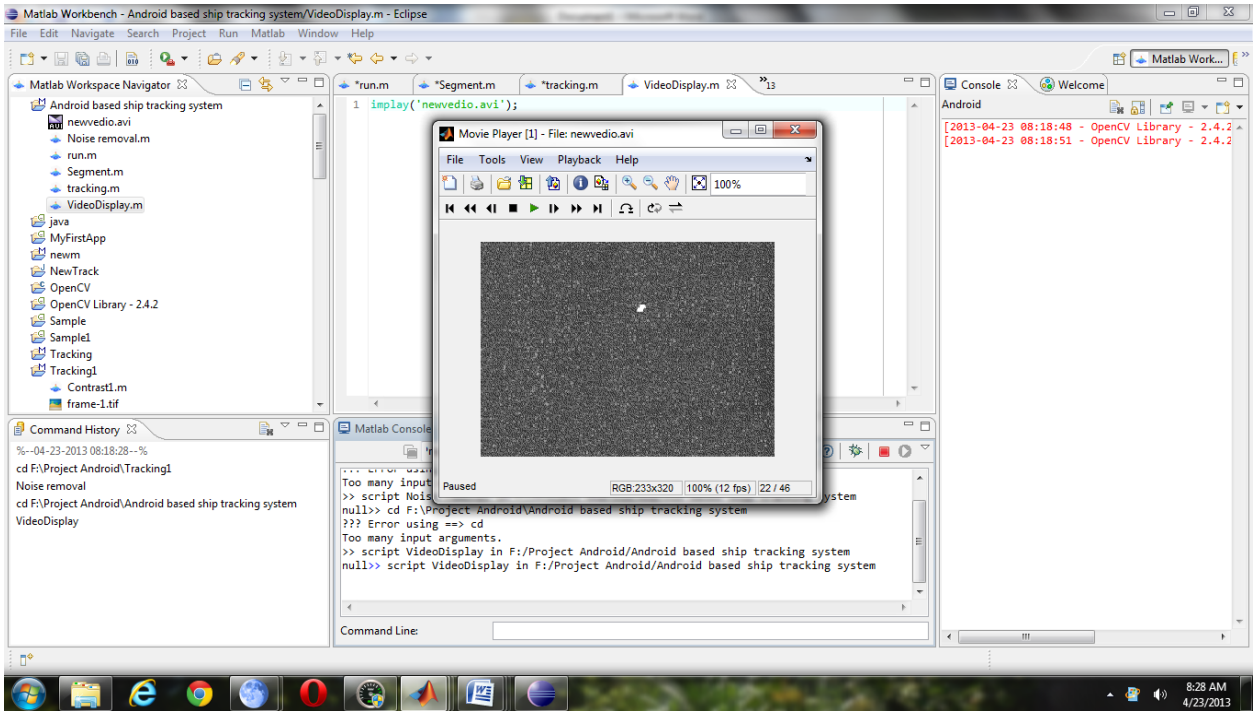
9.3 INPUT SCREENS



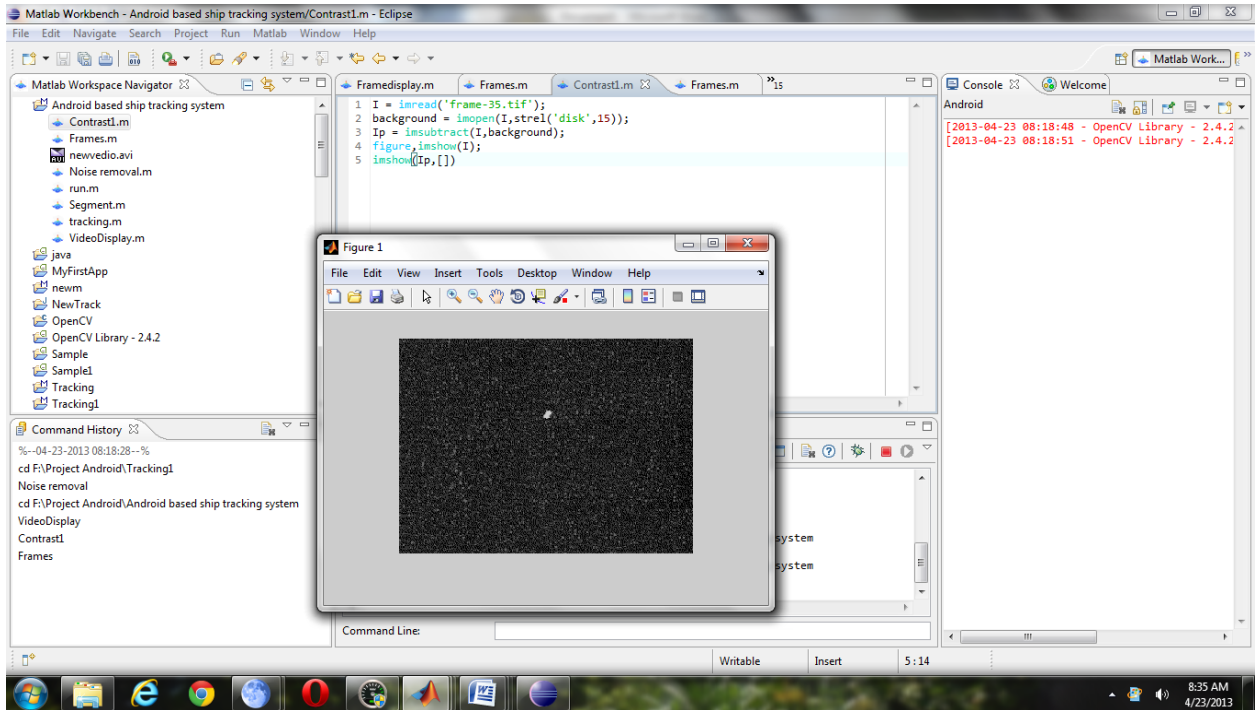
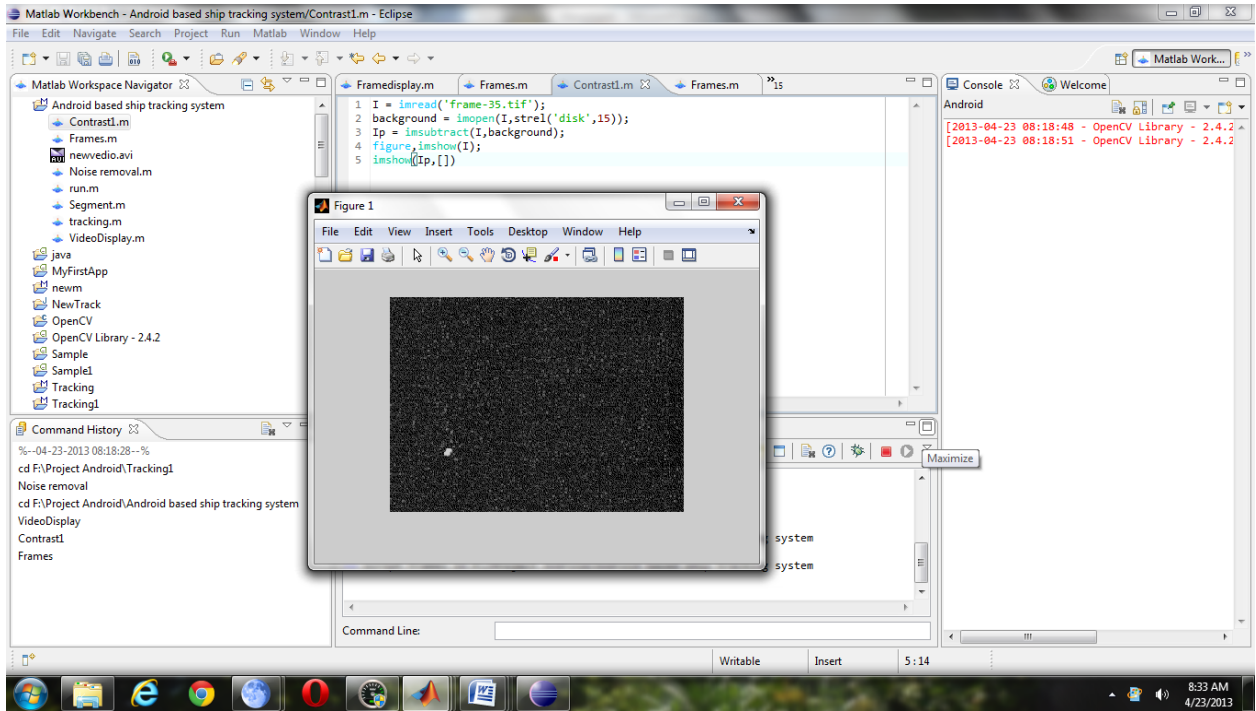
9.4 OUTPUT SCREENS

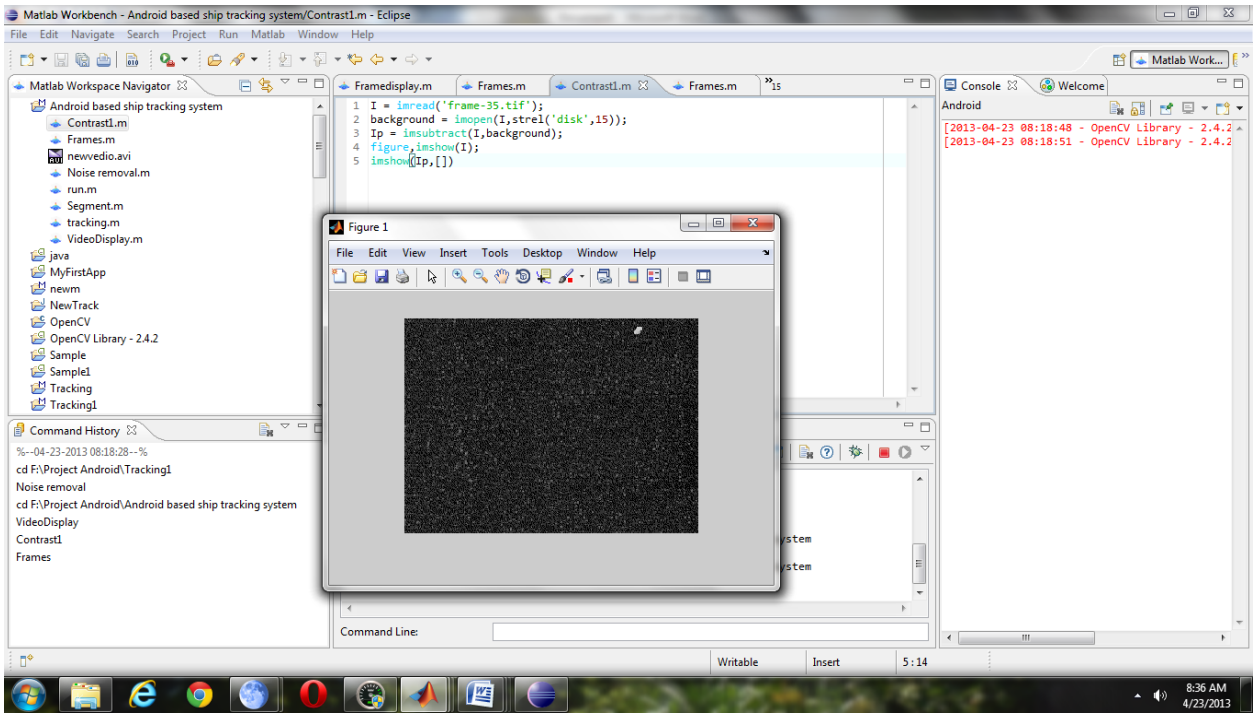
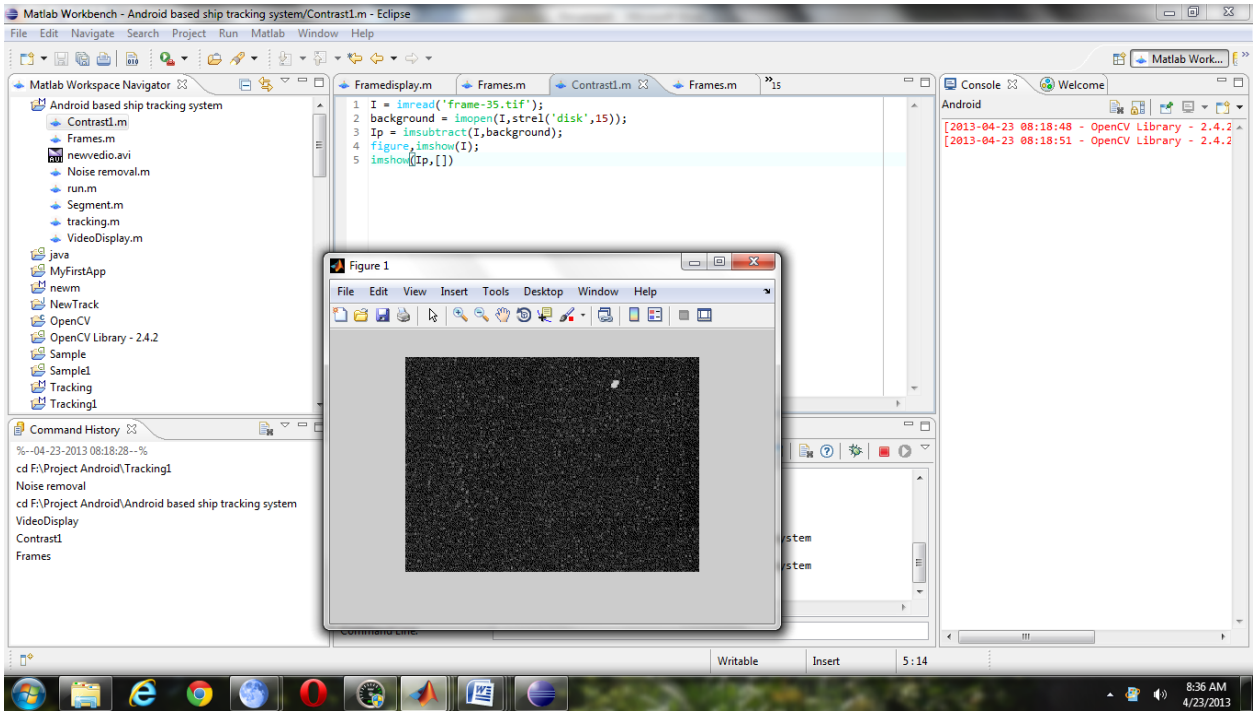
9.4.1 OUTPUT SCREEN FOR VIDEO LOADING AT DIFFERENT FRAMES





9.4.2 OUTPUT SCREEN FOR PREPROCESSING (CONTRAST ENHANCEMENT) AT DIFFERENT FRAMES





Matlab Workbench - Android based ship tracking system/Contrast1.m - Eclipse

File Edit Navigate Search Project Run Matlab Window Help

Matlab Workspace Navigator

- Android based ship tracking system
 - Contrast1.m
 - Frames.m
 - newvedio.avi
 - Noise removal.m
 - run.m
 - Segment.m
 - tracking.m
 - VideoDisplay.m
- java
- MyFirstApp
- newm
- NewTrack
- OpenCV
- OpenCV Library - 2.4.2
- Sample
- Sample1
- Tracking
- Tracking1

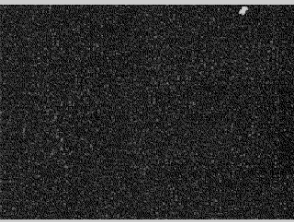
Command History

```
%--04-23-2013 08:18:28--%
cd F:\Project Android\Tracking1
Noise removal
cd F:\Project Android\Android based ship tracking system
VideoDisplay
Contrast1
Frames
```

Framesdisplay.m

```
1 I = imread('frame-35.tif');
2 background = imopen(I, strel('disk', 15));
3 Ip = imsubtract(I, background);
4 figure, imshow(Ip);
5 imshow(Ip, [])
```

Figure 1



Console

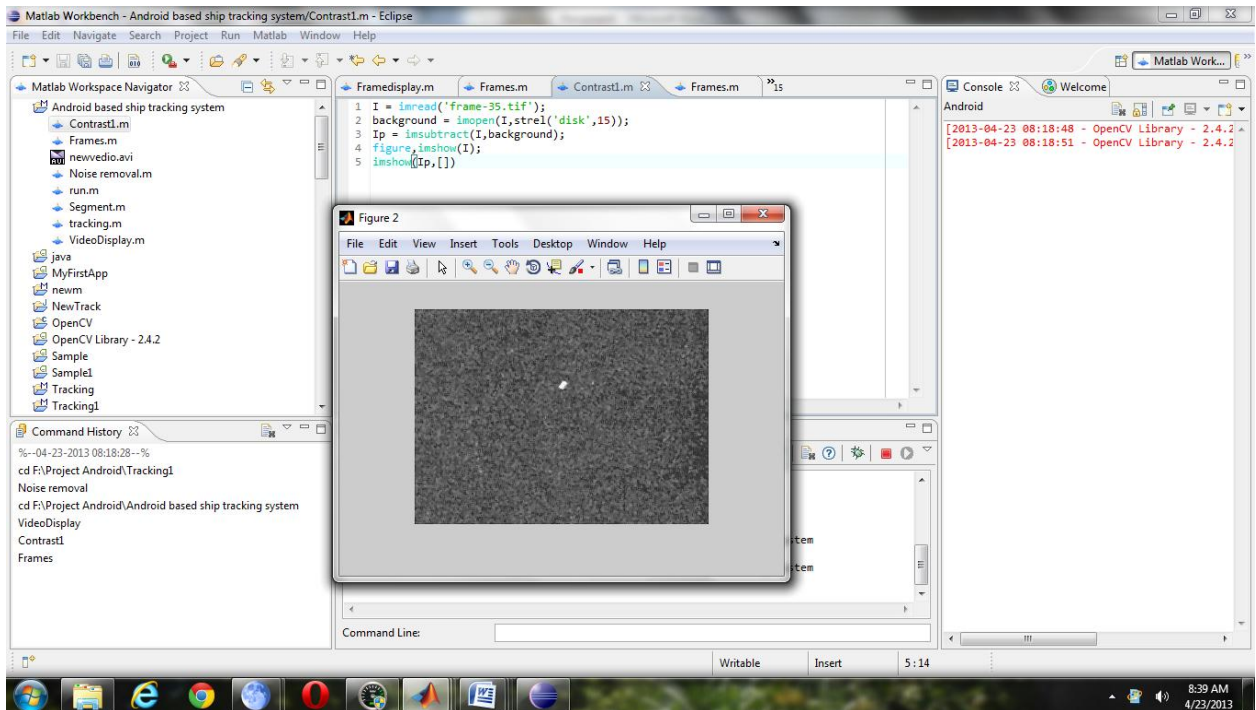
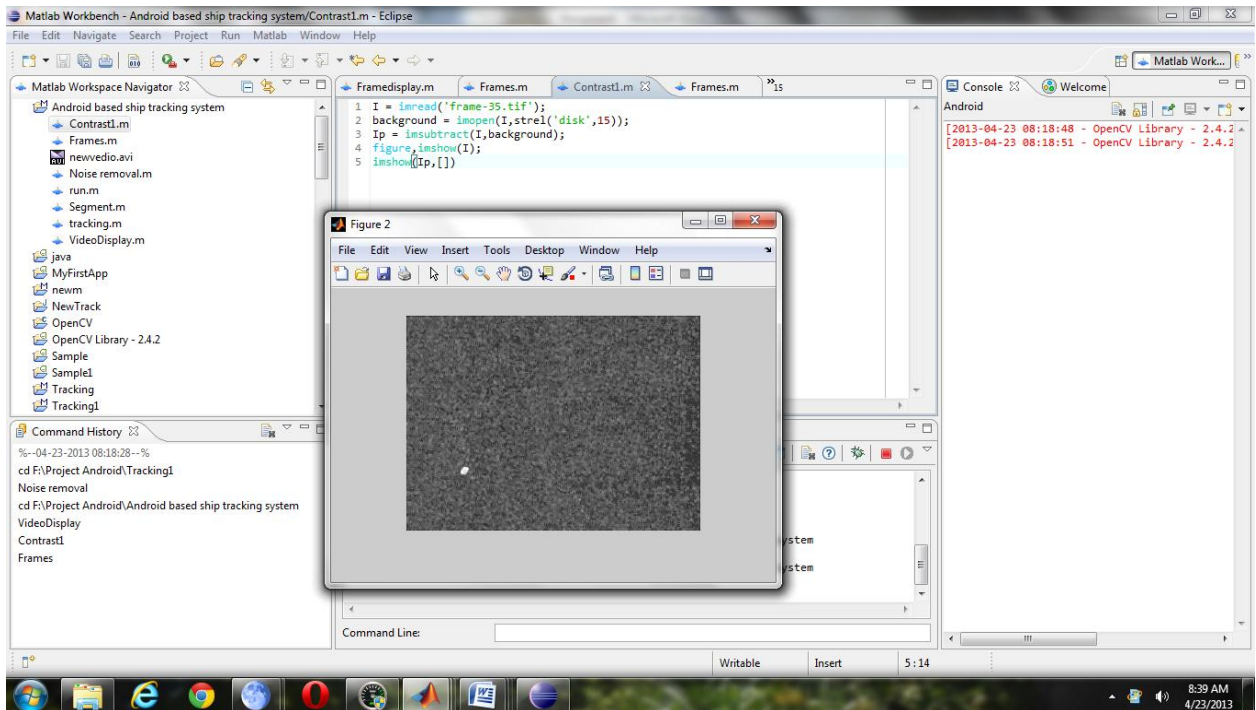
```
Welcome
Android
[2013-04-23 08:18:48 - OpenCV Library - 2.4.2
[2013-04-23 08:18:51 - OpenCV Library - 2.4.2
```

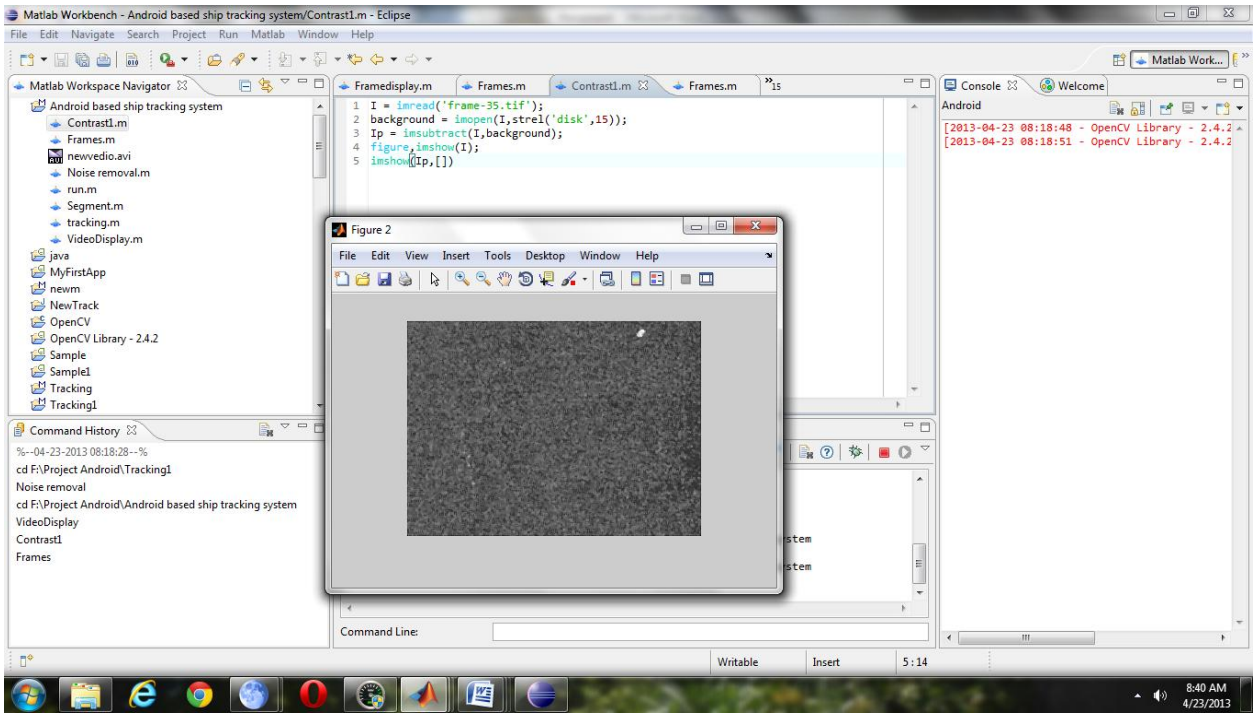
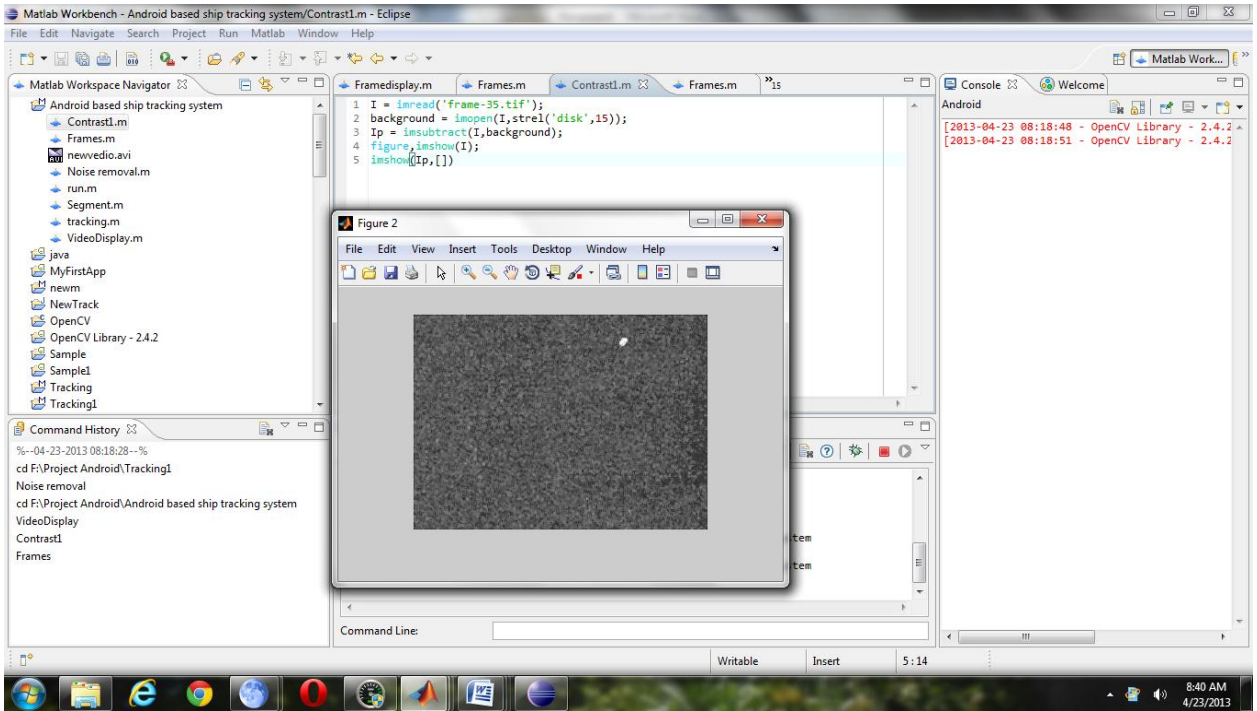
Command Line

Writable Insert 5:14

8:37 AM 4/23/2013

9.4.3 OUTPUT SCREEN FOR PREPROCESSING (NOISE REMOVAL) AT DIFFERENT FRAMES





Matlab Workbench - Android based ship tracking system/Contrast1.m - Eclipse

File Edit Navigate Search Project Run Matlab Window Help

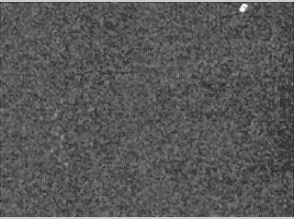
Matlab Workspace Navigator

- Android based ship tracking system
 - Contrast1.m
 - Frames.m
 - newvedio.avi
 - Noise removal.m
 - run.m
 - Segment.m
 - tracking.m
 - VideoDisplay.m
- java
- MyFirstApp
- newm
- NewTrack
- OpenCV
- OpenCV Library - 2.4.2
- Sample
- Sample1
- Tracking
- Tracking1

Command History

```
%--04-23-2013 08:18:28--%
cd F:\Project Android\Tracking1
Noise removal
cd F:\Project Android\Android based ship tracking system
VideoDisplay
Contrast1
Frames
```

Figure 2



Console

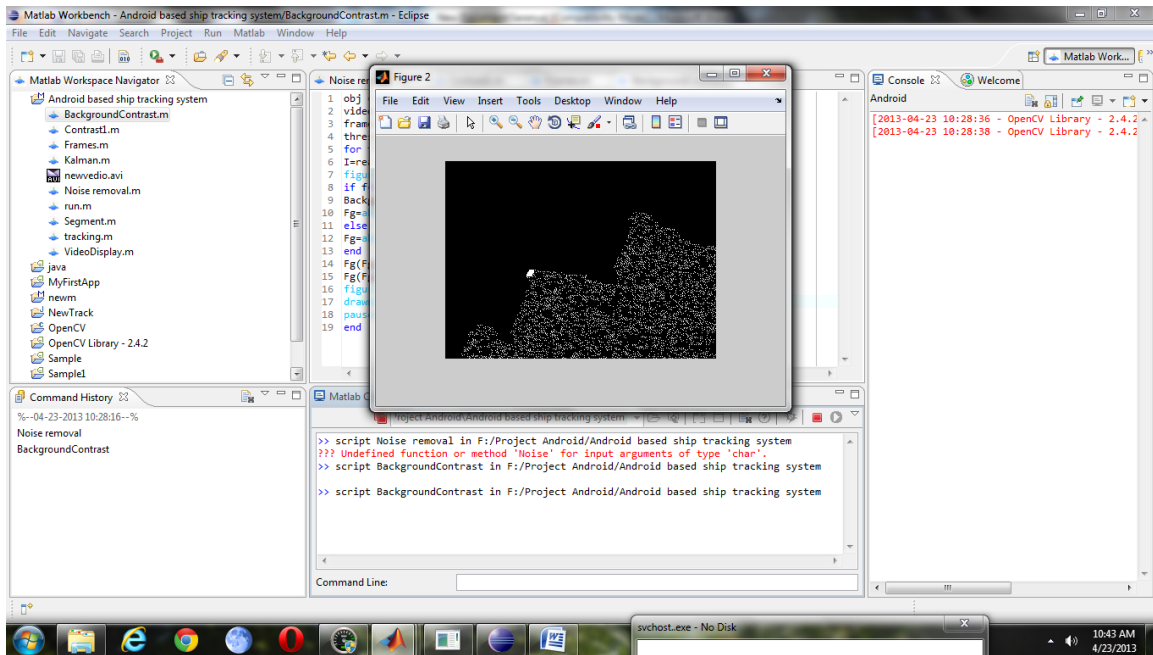
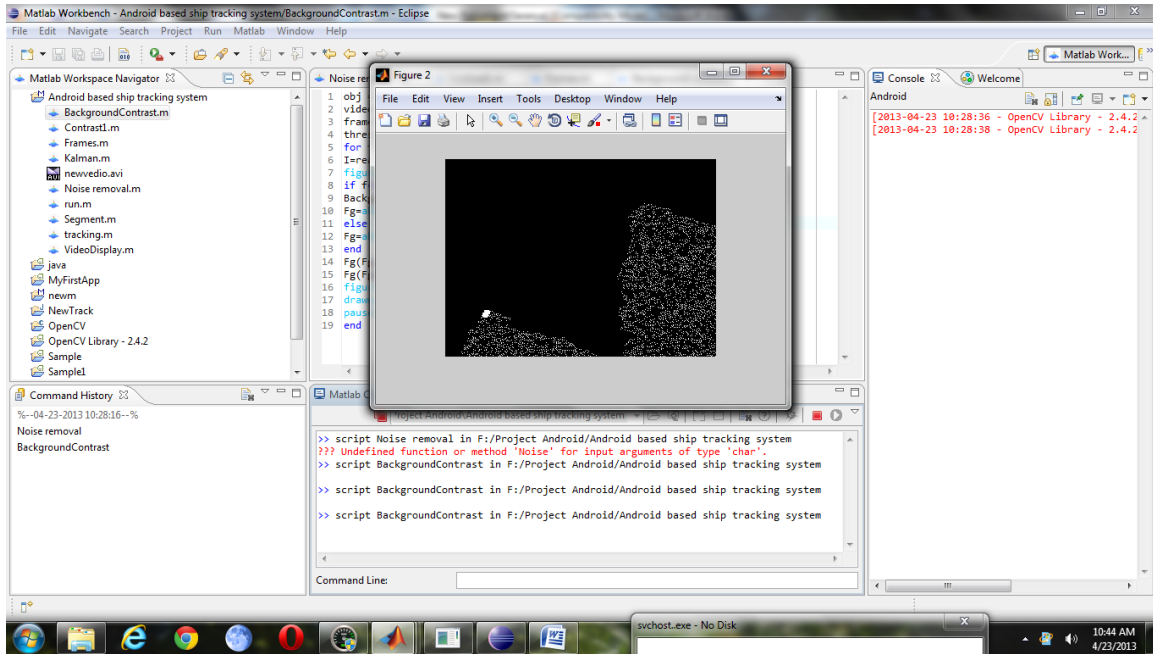
```
Android
[2013-04-23 08:18:48 - OpenCV Library - 2.4.2
[2013-04-23 08:18:51 - OpenCV Library - 2.4.2
```

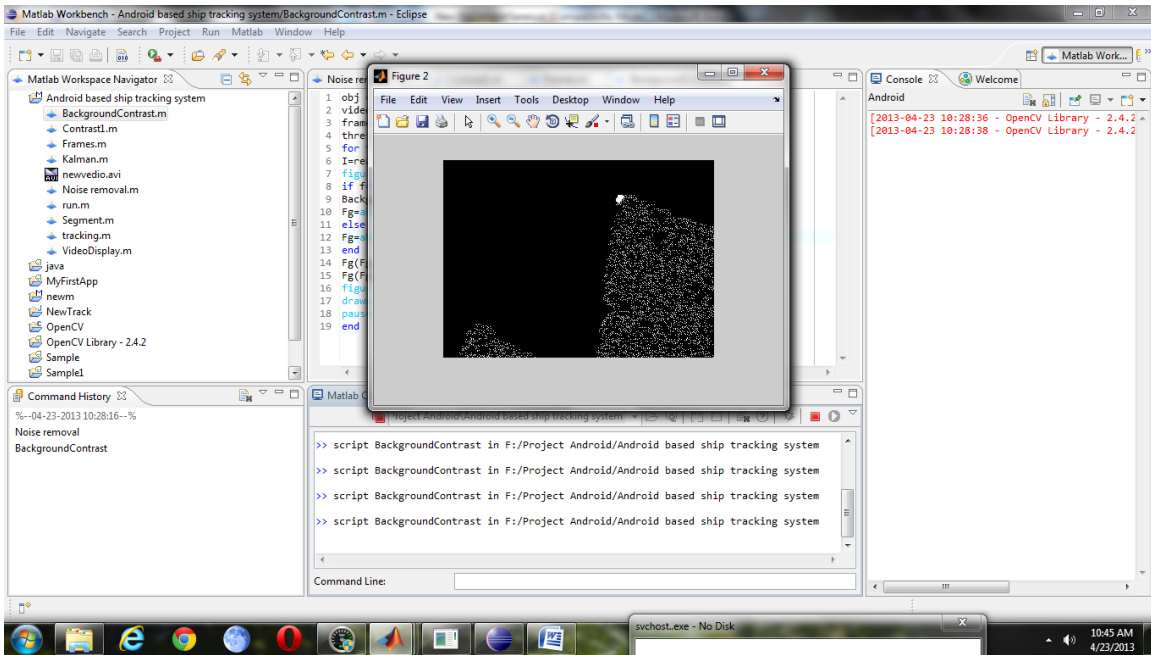
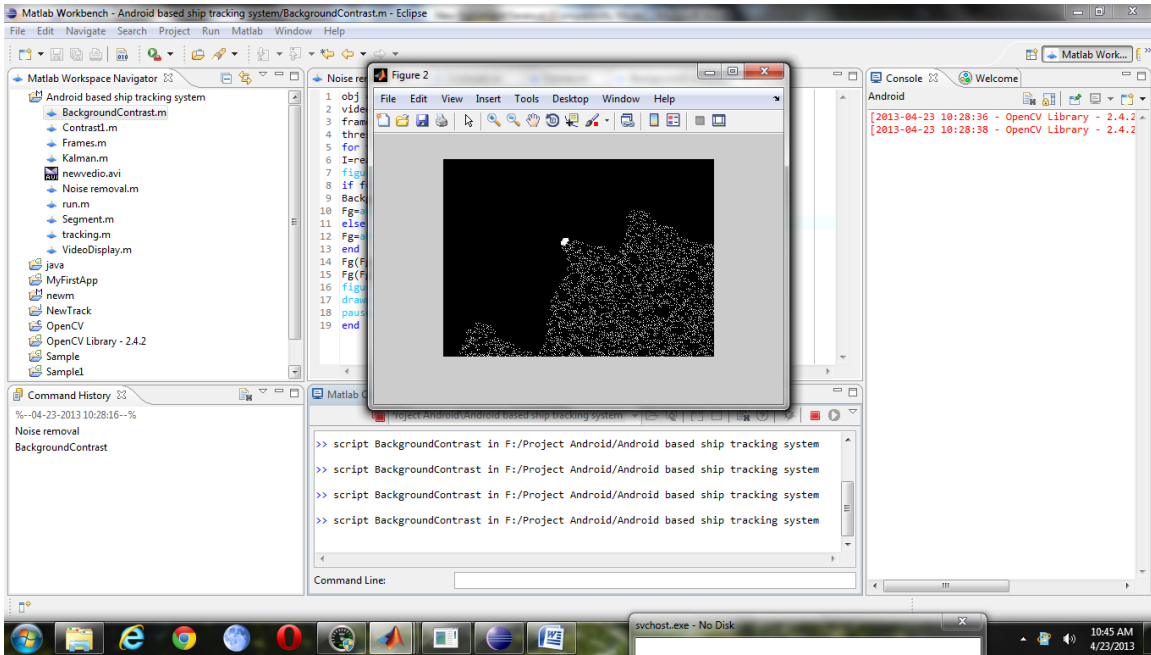
Command Line

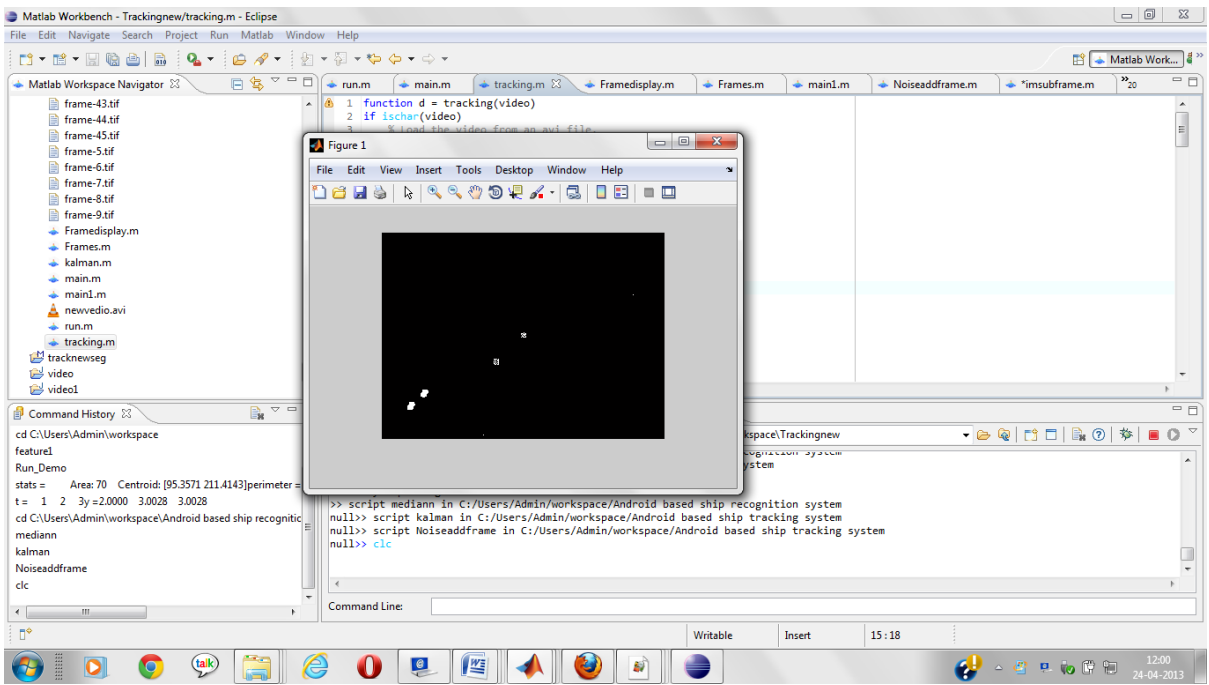
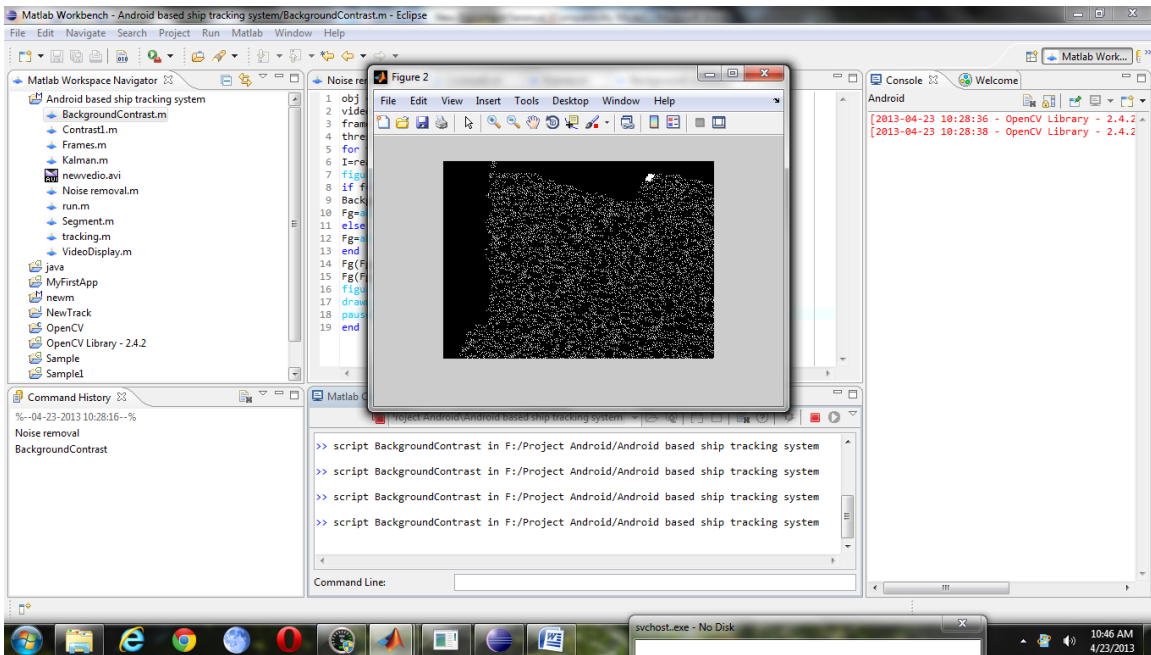
Writable Insert 5:14

8:41 AM
4/23/2013

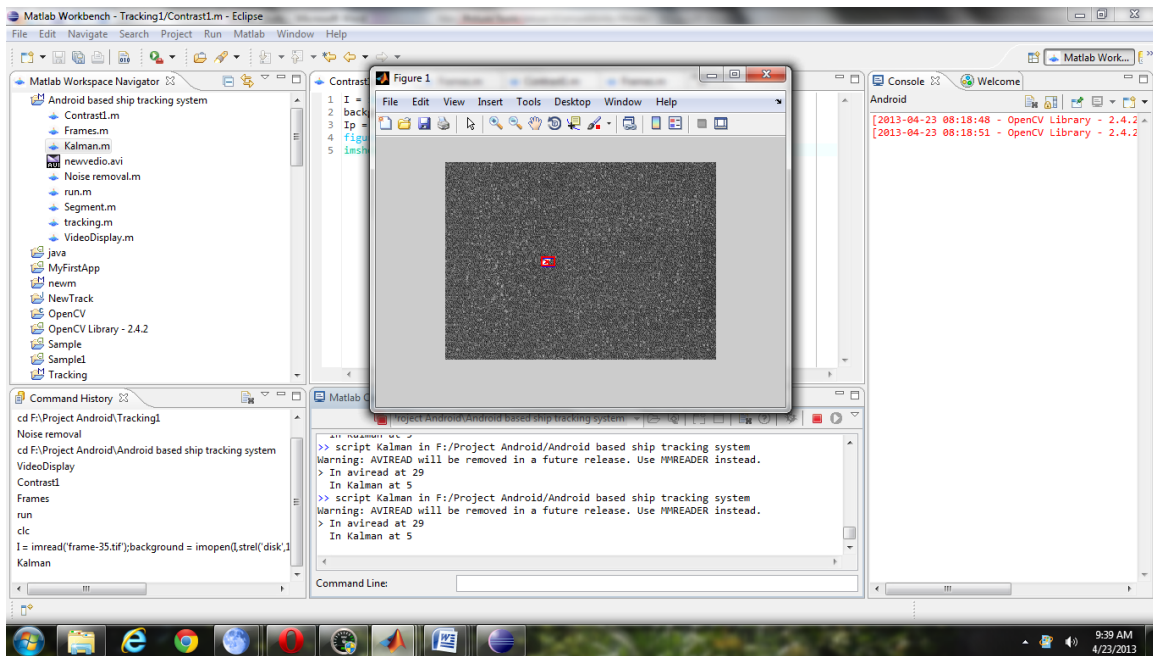
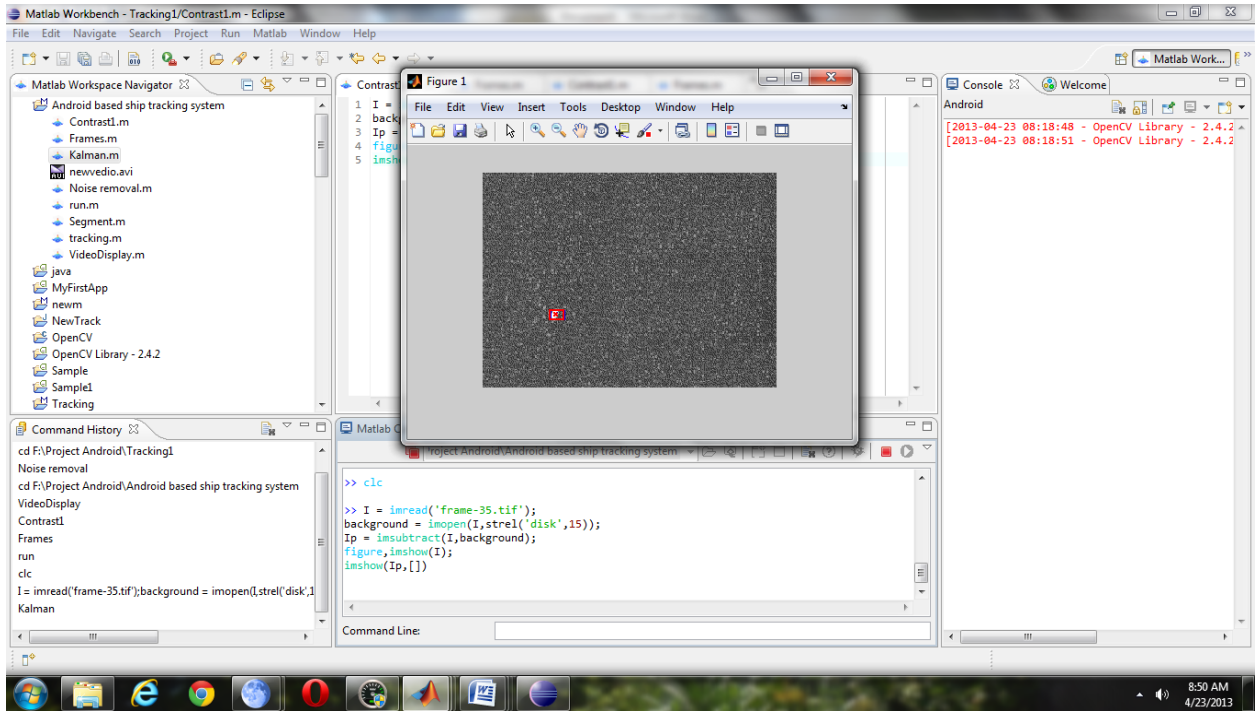
9.4.4 OUTPUT SCREEN FOR SEGMENTATION AT DIFFERENT FRAMES

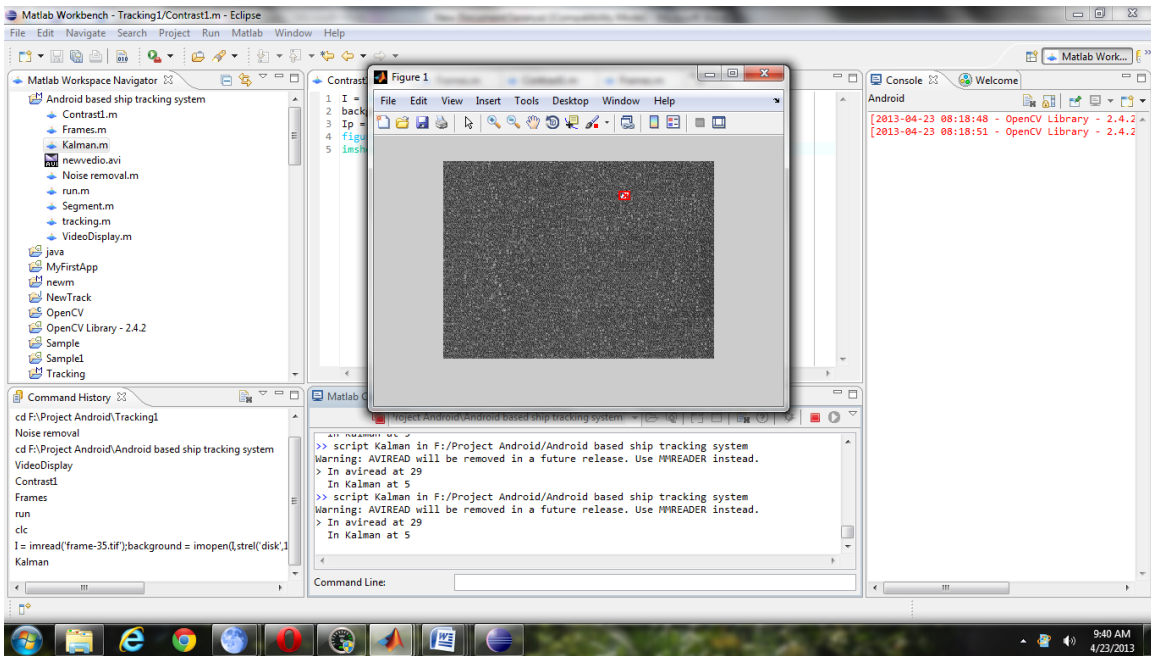
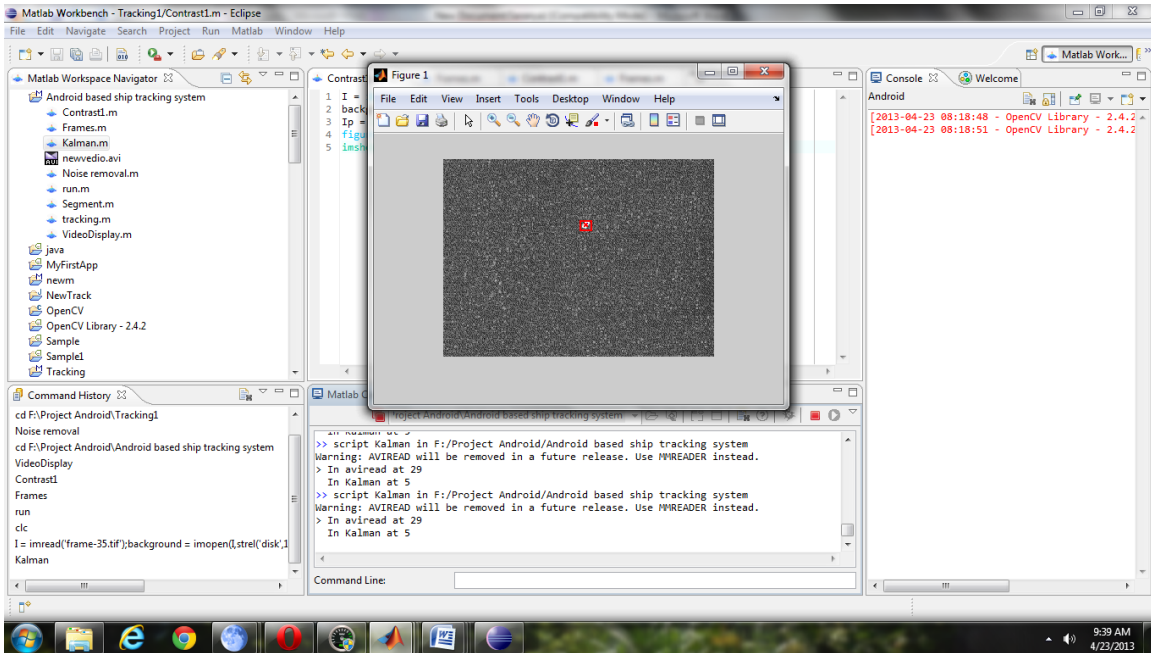


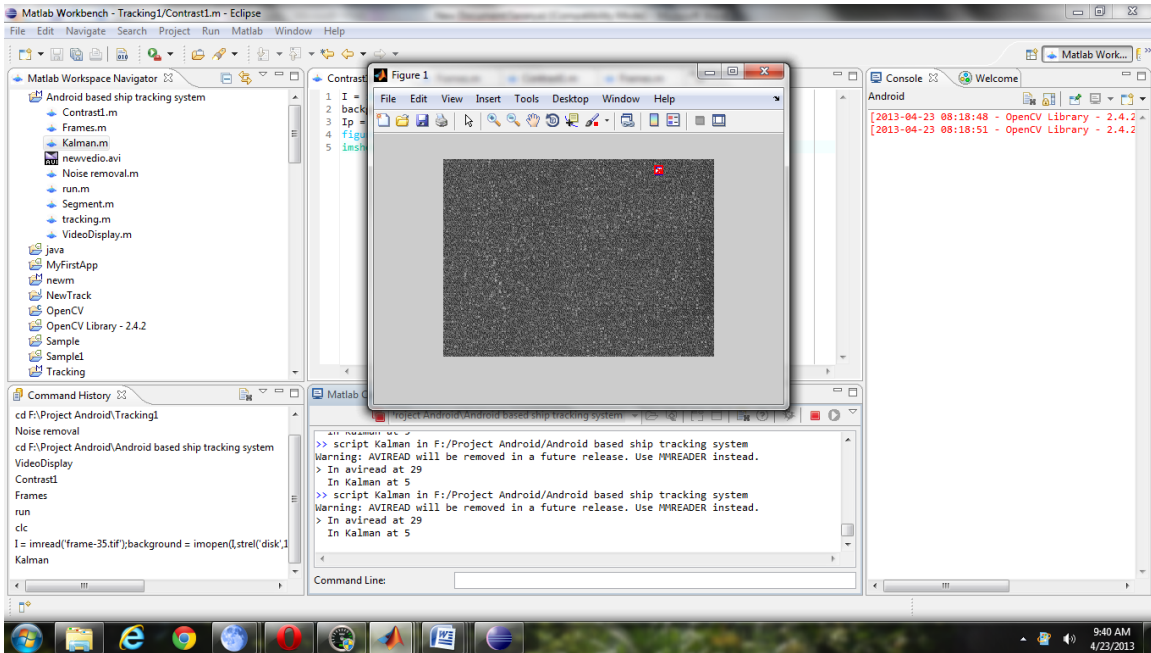




9.4.5 OUTPUT SCREEN FOR KALMAN FILTER BASED TRACKING



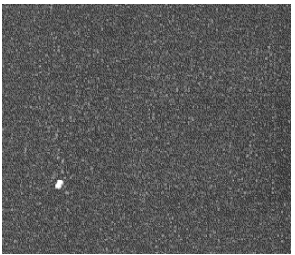
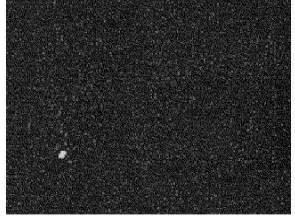
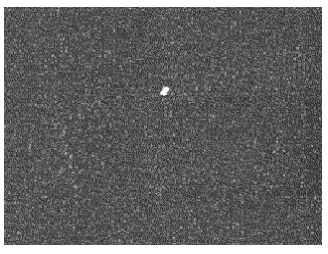
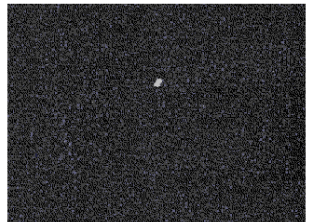


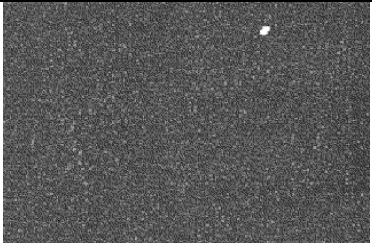
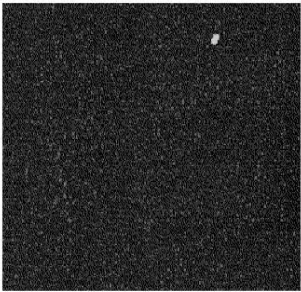
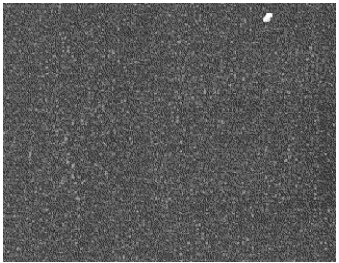
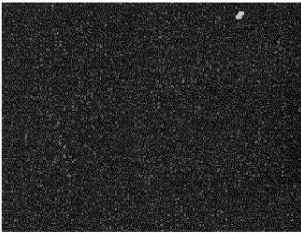
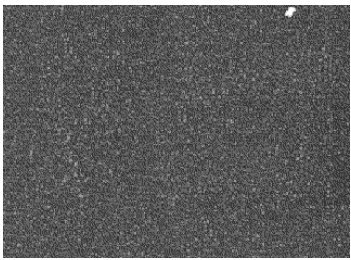
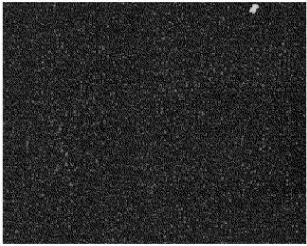


9.5 RESULTS

9.5.1 INPUT FRAMES AND PREPROCESSED FRAMES

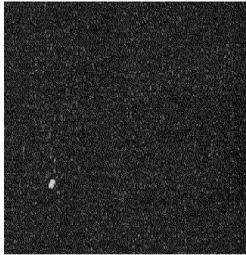
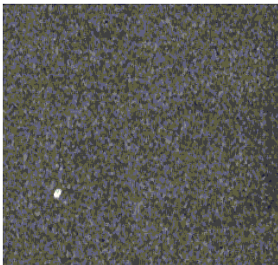
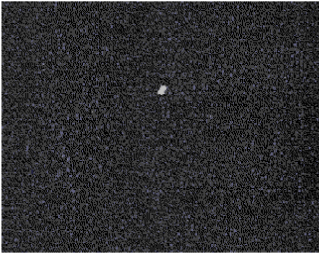
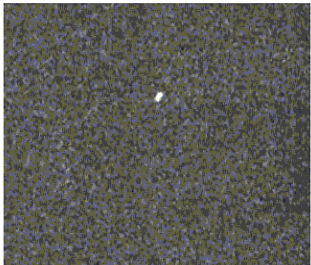
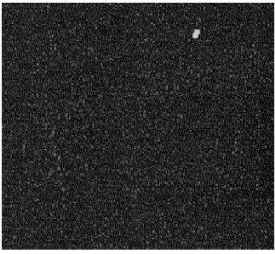
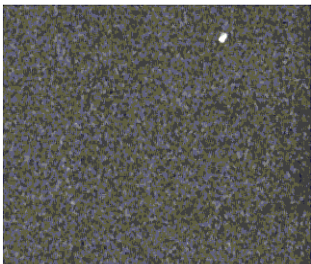
CONTRAST ENHANCEMENT

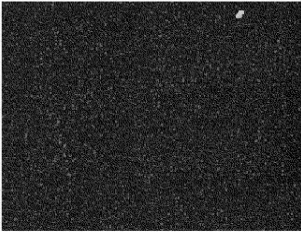
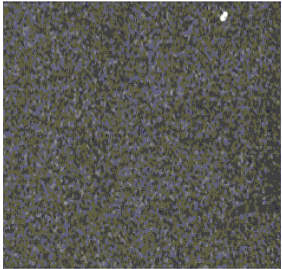
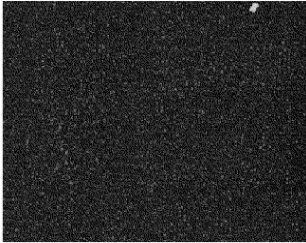
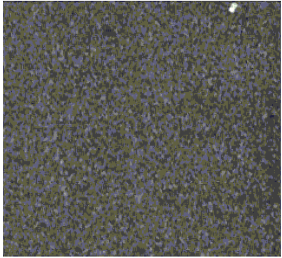
FRAME NUMBERS	INPUT VIDEO FRAMES	CONTRAST ENHANCED FRAMES
1		
2		
3		

		
4		
5		

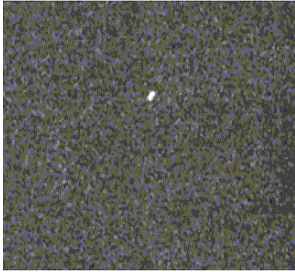
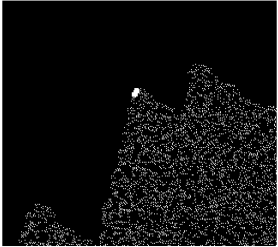
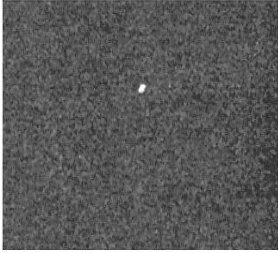
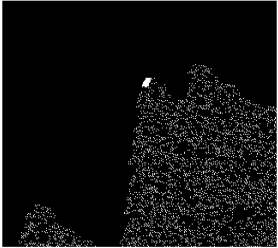
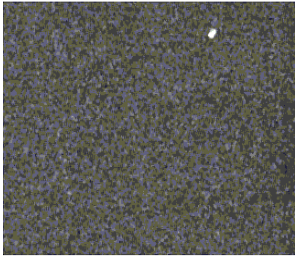
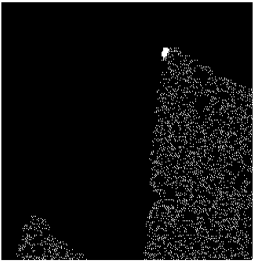
9.5.2 INPUT FRAMES AND PREPROCESSED FRAMES

NOISE REMOVAL

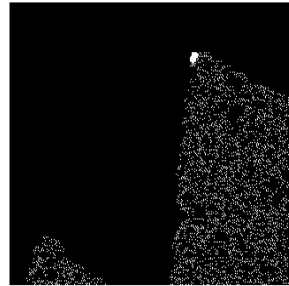
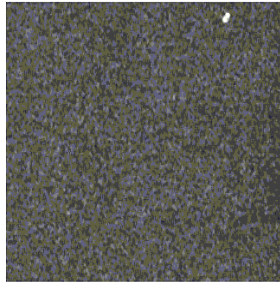
FRAME NUMBERS	INPUT VIDEO FRAMES	NOISE REMOVED FRAMES
1		
2		
3		

4		
5		

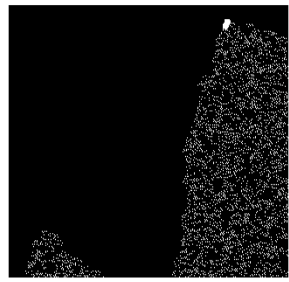
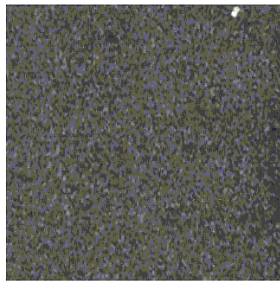
9.5.3 INPUT FRAMES AND SEGMENTED FRAMES

FRAME NUMBERS	INPUT VIDEO FRAMES	SEGMENTATION
1		
2		
3		

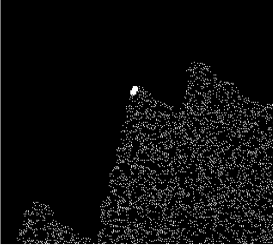
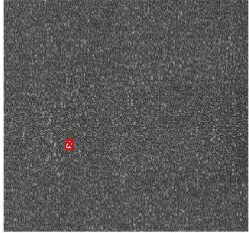
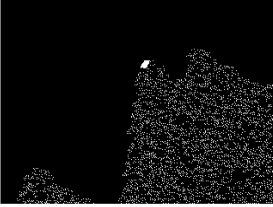
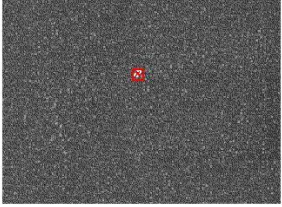
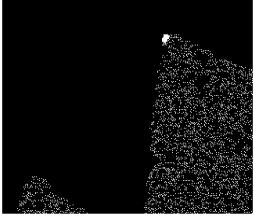
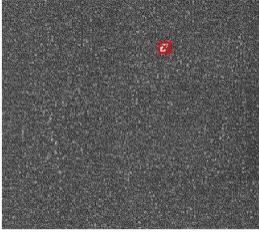
4



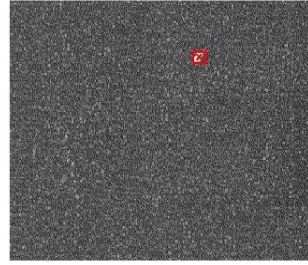
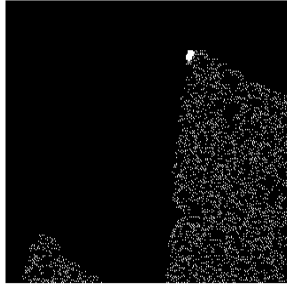
5



9.5.4 INPUT FRAMES AND TRACKING

FRAME NUMBER	INPUT VIDEO FRAMES	TRACKING
1		
2		
3		

4



5

