# FORAGING ROBOTS USING SWARM INTELLIGENCE

**A PROJECT REPORT**

***Submitted by***



| | |
|---|---|
| **SARANYA.S** | **Reg. No.:1110107077** |
| **SARANYA.S** | **Reg. No.:1110107078** |
| **VISHNU DHARSANA.R** | **Reg. No.:1110107116** |



***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

**IN**

**ELECTRONICS AND COMMUNICATION ENGINEERING**



**KUMARAGURU COLLEGE OF TECHNOLOGY**

**COIMBATORE-641049**

**(An Autonomous Institution Affiliated to Anna University, Chennai)**



**APRIL 2015**

# KUMARAGURU COLLEGE OF TECHNOLOGY

# COIMBATORE-641049

**(An Autonomous Institution Affiliated to Anna University, Chennai)**

## BONAFIDE CERTIFICATE

Certified that this project report titled **"FORAGING ROBOTS USING SWARM INTELLIGENCE"** is the bonafied work of "**SARANYA.S, SARANYA.S, VISHNU DHARSANA .R"** who carried out the project work under my supervision.

| | |
|---|---|
| **SIGNATURE** | **SIGNATURE** |
| Ms. M.Shanthi, M.S.,(Ph.D)., | Dr. Rajeswari Mariappan M.E., Ph.D., |
| Associate Professor/E.C.E | **HEAD OF THE DEPARTMENT** |
| Kumaraguru College of Technology | Electronics & Communication Engineering |
| Coimbatore. | Kumaraguru College of Technology |
| | Coimbatore. |

The candidates with Register numbers 1110107077, 1110107078 and 1110107116 are examined by us in the project viva-voce examination held on …………………….

**INTERNAL EXAMINER** **EXTERNAL EXAMINER**

## ACKNOWLEDGEMENT

First we would like to express our praise and gratitude to the Lord, who has showered his grace and blessing enabling us to complete this project in an excellent manner. He has made all things in beautiful in his time.

We express our sincere thanks to our beloved Joint Correspondent, **Shri. Shankar Vanavarayar** for his kind support and for providing necessary facilities to carry out the project work.

We would like to express our sincere thanks to our beloved Principal **Dr. R. S. Kumar M.E., Ph. D.,** who encouraged us with his valuable thoughts.

We would like to express our sincere thanks and deep sense of gratitude to our HOD, **Dr. Rajeswari Mariappan M.E., Ph. D.,** for her valuable suggestions and encouragement which paved way for the successful completion of the project.

We are greatly privileged to express our deep sense of gratitude to the Project Coordinator **Ms. A. Kalaiselvi M. E. , (Ph. D)**, Assistant Professor, for her continuous support throughout the course.

In particular, We wish to thank and express our everlasting gratitude to the Supervisor **Ms. M. Shanthi M.S.,(Ph.D),** Associate Professor for her expert counselling in each and every steps of project work and we wish to convey our deep sense of gratitude to all teaching and non-teaching staff members of ECE Department for their help and cooperation.

Finally, we thank our parents and our family members for giving us the moral support in all of our activities and our dear friends who helped us to endure our difficult times with their unfailing support and warm wishes.

# ABSTRACT

Inspired by the collective behaviour observed in natural insects, Swarm Robotics is a new approach of design to demonstrate swarm intelligence in a large group of robots. In such a robotic system individual robot has only limited capabilities in terms of sensing, computation, and communication but its behaviour can be designed such that a desired collective behaviour emerges from the local interactions among robots and between the robots and the environment. The attention to swarm robotics has increased recently because of beneficial features demonstrated in such systems, such as higher group efficiency, the robustness against failures of individuals, flexibility to adapt to changes in the environment and scalability over a wide range of group sizes. In this paper we present an adaptive algorithm to regulate the behaviour of individual robot performing foraging task. Through the interactions between robots, a desired division of labour can be achieved in group level. Robot group also demonstrates the ability to improve energy efficiency and its potential robustness in different environments.

# TABLE OF CONTENTS

## LIST OF FIGURES

# 1. INTRODUCTION

Swarm robotics has been defined as “a novel approach to the coordination of large numbers of robots” and as “the study of how large numbers of relatively simple physically embodied agents can be designed such that a desired collective behavior emerges from the local interactions among agents and between the agents and the Environment. The main characteristics of a swarm robotics system are (i)robots are autonomous (ii) robots are suitable for most environments and can act to modify it (iii)robots’ sensing and communication capabilities are local (iv) robots do not have access to centralized control and/or to global knowledge (v)robots cooperate to tackle a given task. Swarm engineering is the systematic application of scientific and technical knowledge to model and specify requirements, design, realize, verify, validate, operate and maintain a swarm intelligence system. In this project, we use these characteristics to discriminate between the works that belong to swarm robotics from those that belong to other multi-robot approaches. The main inspiration for swarm robotics comes from the observation of social animals. Ants, bees, birds and fish are some examples of how simple individuals can become successful when they gather in groups. The interest towards social animals stems from the fact that they exhibit a sort of swarm intelligence. In particular, the behavior of groups of social animals appear to be robust, scalable and flexible. In particular, swarm robotics systems are meant to be robust, scalable and flexible.

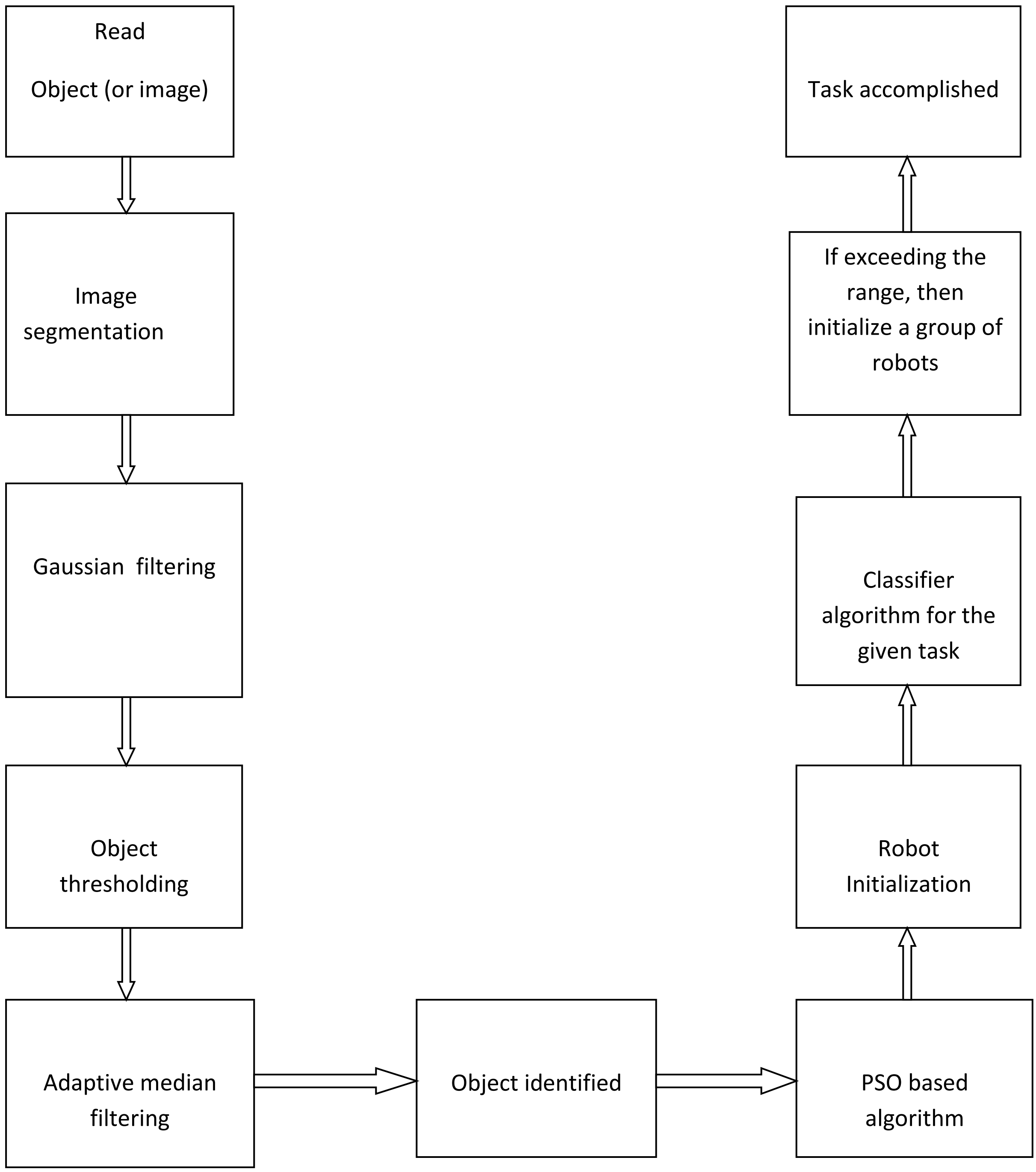



**Figure 1.1:** Process flow of the project

# 2. IMAGE IDENTIFICATION

## 2.1 IMAGE

### 2.1.1 IMAGE DEFINITION

An image is an array or a matrix of square pixels(picture elements) arranged in columns and rows. An image (from Latinn: imago) is an artifact, for example a two-dimensional picture, that has a similar appearance to some subject-usually a physical object or a person.

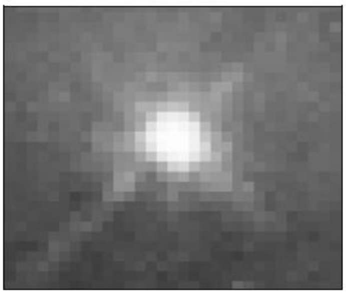


**Figure 2.1**: An image – an array or a matrix of pixels arranged in columns and rows

### 2.1.2 IMAGE CHARACTERISTICS

Images may be two-dimensional, such as a photograph, screen display, and as well as a three-dimensional, such as a statue or hologram. They may be captured by optical devices—such as cameras, mirrors, lenses, telescopes, microscopes, etc. and natural objects and phenomena, such as the human eye or water surfaces. The word image is also used in the broader sense of any two-dimensional figure such as a map, a graph, a pie chart, or an abstract painting. In this wider sense, images can also be rendered manually, such as by drawing, painting, carving, rendered automatically by printing or computer graphics technology, or developed by a combination of methods, especially in a pseudo-photograph.

### 2.1.3 PIXEL

Image processing is a subset of the electronic domain wherein the image is converted to an array of small integers, called *pixels*, representing a physical quantity such as scene radiance, stored in a digital memory and processed by computer or other digital hardware. Suppose we take an image, a photo, say. For the moment, lets make things easy and suppose the photo is black and white (that is, lots of shades of grey), so no colour. We may consider this image as being a two dimensional function, where the function values give the brightness of the image at any given point, as shown in figure 1. We may assume that in such an image brightness values can be any real numbers in the range 0.0 (black) to 1.0 (white). The ranges of x and y will clearly depend on the image, but they can take all real values between their minima and maxima.

A digital image differs from a photo in that the x, y, and f(x,y)values are all discrete. Usuallythey take on only integer values, so the image shown in figure 1.2 will have x and y ranging from 1to 256 each, and the brightness values also ranging from 0 (black) to 255 (white). A digital image can be considered as a large array of discrete dots, each of which has a brightness associated with it. These dots are called picture elements, or more simply pixels. The pixels surrounding a given pixel constitute its neighbourhood. A neighbourhood can be characterized by its shape in the same way as a matrix: we can speak of a 3*3 neighbourhood, or of a 5*7 neighbourhood. Except in very special circumstances, neighbourhoods have odd numbers of rows and columns; this ensures that the current pixel is in the centre of the neighbourhood. An example of a neighbourhood is given in figure 1.3. If a neighbourhood has an even number of rows or columns (or both), it may be necessary to specify which pixel in the neighbourhood is the "current pixel".

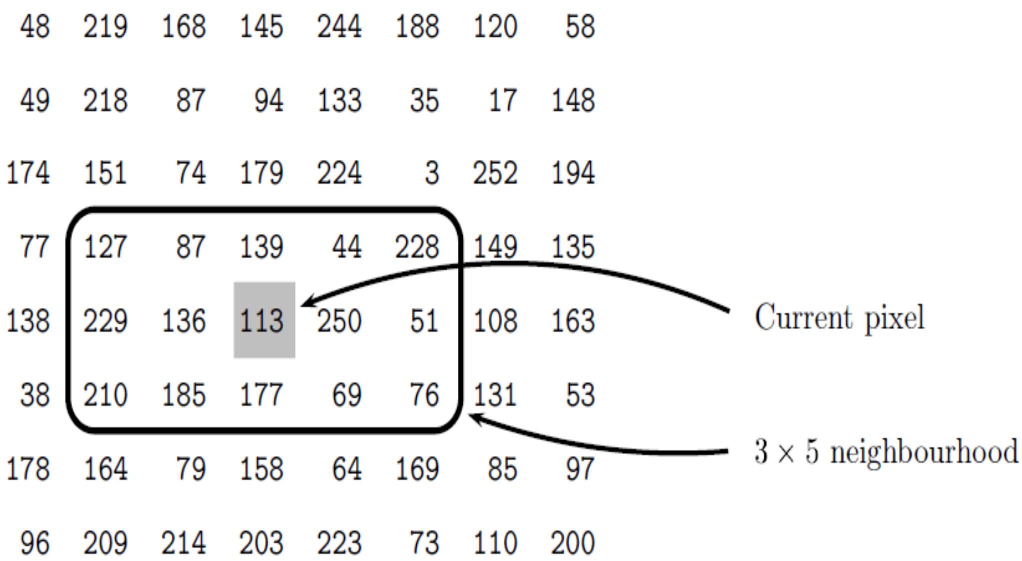



**Figure2.2:** Pixel representation

2.1.4 IMAGE RESOLUTION

Pixels transform into inches through what is called “resolution,” -- the number of pixels per square inch on a computer. Resolution allows you to transform pixels into inches and back again.Two resolution definitions are often used in place of one another. Pixel resolution is the size (in bytes) of your image or its appearance on a computer screen. This number is tied directly to how big your image is on your hard drive. The byte-size of the image file is directly proportional to the pixel count and its size on your computer screen, which simply displays all the pixels in a fixed one-to-one grid.

2.1.5 IMAGE BRIGHTNESS

‘Brightness’ is a problem word, used in most descriptions of image quality. Often, if an image looks bad we will say that it’s not ‘bright’, when it merely lacks contrast. So what does ‘brightness’ actually mean? Brightness actually describes how we experience light and not ‘how it is’. If we are going to describe light and ‘brightness’ properly, there are two essential terms namely luminance and illuminance. Luminance is the light we see: reflected or radiating from objects. It is Measured in candela per square metre (cd/m2 and the same as ‘nit’). Illuminance is light we can’t see directly, it is so called ambient light, light passing through air, actually invisible until it reflects off an object (which we then see as luminance). It is measured in lux (lx or lm/m2).

### 2.1.6 CONTRAST

Contrast refers to the difference between black and white levels in images, whether on a flat panel display or a projection screen. Without good contrast, images appear to lack ‘brightness’, colour and definition. The image contrast ratio refers to the difference between the luminance of the white part of an image, divided by the black part. So if the white part is one hundred times brighter than the black part, it will be 100:1, and so on.

## 2.2 GRAY THRESHOLD ALGORITHM

In many applications of image processing, the gray levels of pixels belonging to the object are quite different from the gray levels of the pixels belonging to the background. Image thresholding is a well known image segmentation procedure extensively attempted to obtain binary image from the gray level image. Here, histogram based bi–level and multi-level segmentation is proposed for gray scale images. The optimal thresholds are attained by maximizing Otsu's between class variance function.

### 2.2.1 BASIC CONCEPTS

For gray scale images, thresholding is widely considered to extract key features from input image. The main objective is to enhance the key feature of an image using the best possible bi-level as well as the multilevel threshold. To make thresholding completely automated, it is necessary for the computer to automatically select the threshold categorize thresholding methods into the following six groups based on the information the algorithm manipulates:

- **Histogram shape-**based methods, where, for example, the peaks, valleys and curvatures of the smoothed histogram are analyzed.
- **Clustering-**based methods, where the gray-level samples are clustered in two parts as background and foreground (object), or alternately are modeled as a mixture of two Gaussians.
- **Entropy**-based methods result in algorithms that use the entropy of the foreground and background regions, the cross-entropy between the original and binarized image, etc.
- **Object Attribute**-based methods search a measure of similarity between the gray-level and the binarized images, such as fuzzy shape similarity, edge coincidence, etc.

- **Spatial** methods [that] use higher-order probability distribution and/or correlation between pixels
- **Local** methods adapt the threshold value on each pixel to the local image characteristics. In these methods, a different T is selected for each pixel in the image.

### 2.2.2 METHODOLOGY

We propose a new Threshold based on Multi-objective Optimization method that combines the flexibility of multi-objective fitness functions with the power of BPSO for searching vast combinatorial state spaces. The idea was inspired from the segmentation problematic:

1) There does not exist any thresholding criterion that is capable to produce an optimal thresholding result for all images. The use of non-Pareto multi-objective optimization aims to obtain good thresholding results independently of the image.
2) Finding the "optimum" number of thresholds, in a whole gray-level range, is usually a challenge since it requires a priori knowledge. However, despite the amount of research in this area, the outcome is still unsatisfactory.

### 2.2.3 IMPLEMENTATION

To solve our multi-objective problem, a method that consists in using a set of subswarms S ={Ss1,…, Ssp,…, SsNc}; Ssp is the sub-swarm p, and Nc is the number of sub-swarm and it represents the number of criteria is used. Each sub-swarm Ssp is valued by using an algorithm binary particle swarm optimisation that searches the optimal thresholds, by optimizing one of objective functions of the problem (thresholding criteria) fp , which uses the gray-level thresholds as parameters. It starts with large number initial thresholds (gray-level range of pixels in the given image). Then, these thresholds are dynamically refined to improve the value of the objective function. The different sub-swarms communicate between them through the exchange of their better position by using the uniformity measure.

### 2.2.3.1: SEGMENTATION CRITERIA

In this approach, we use three threshold criteria and a selection operator of the best thresholds. The threshold criteria can be described as follows: let there be N pixels in a given image, with gray-level range over [0,…..,L] and ni denote the occurrence of gray-level i, giving a probability of gray-level i.

### 2.2.3.2: Selection Criterion

The uniformity measure U is used for evaluating the quality of thresholded image and eventually to select the best thresholds.

## 2.3 FILTERING METHODS

We use two main types of filtering here in our project:

1) Gaussian Filtering
2) Median Filtering

### 2.3.1 GAUSSIAN FILTERING

The Gaussian function is used in numerous research areas :

- It defines a probability distribution for noise or data

- It is a smoothing operator

- It is used in mathematics

In probabilistic terms, it describes 100% of the possible values of any given space when varying from negative to positive values

-Gauss function is never equal to zero.

-It is a symmetric function

The Gaussian filter works by using the 2D distribution as a point-spread function. This is achieved by convolving the 2D Gaussian distribution function with the image. We need to produce a discrete approximation to the Gaussian function. This theoretically requires an infinitely large convolution kernel, as the Gaussian distribution is non-zero everywhere. Fortunately the distribution has approached very close to zero at about three standard deviations from the mean. 99% of the distribution falls within 3 standard deviations. This means we can normally limit the kernel size to contain only values within three standard deviations of the mean.

### 2.3.2 MEDIAN FILTERING

Median filtering does the following functions on the selected image :

1) Remove impulse noise

2) Smoothing of other noise

3) Reduce distortion, like excessive thinning or thickening of object boundaries.

In many different kinds of digital image processing, the basic operation is as follows: at each pixel in a digital image we place a neighborhood around that point, analyze the values of all the pixels in the neighborhood according to some algorithm, and then replace the original pixel's value with one based on the analysis performed on the pixels in the neighborhood. The neighborhood then moves successively over every pixel in the image, repeating the process.

Median filtering follows this basic prescription. The median filter is normally used to reduce noise in an image, somewhat like the mean filter. However, it often does a better job than the mean filter of preserving useful detail in the image. This class of filter belongs to the class of edge preserving smoothing filters which are non-linear filters. This means that for two images *A(x)* and *B(x)*:

$$median[A(x) + B(x)] \neq median[A(x)] + median[B(x)]$$

These filters smooths the data while keeping the small and sharp details. The median is just the middle value of all the values of the pixels in the neighborhood. Note that this is not the same as the average (or mean); instead, the median has half the values in the neighborhood larger and half smaller. The median is a stronger "central indicator" than the average. In particular, the median is hardly affected by a small number of discrepant values among the pixels in the neighborhood. Consequently, median filtering is very effective at removing various kinds of noise. Figure  illustrates an example of median filtering.

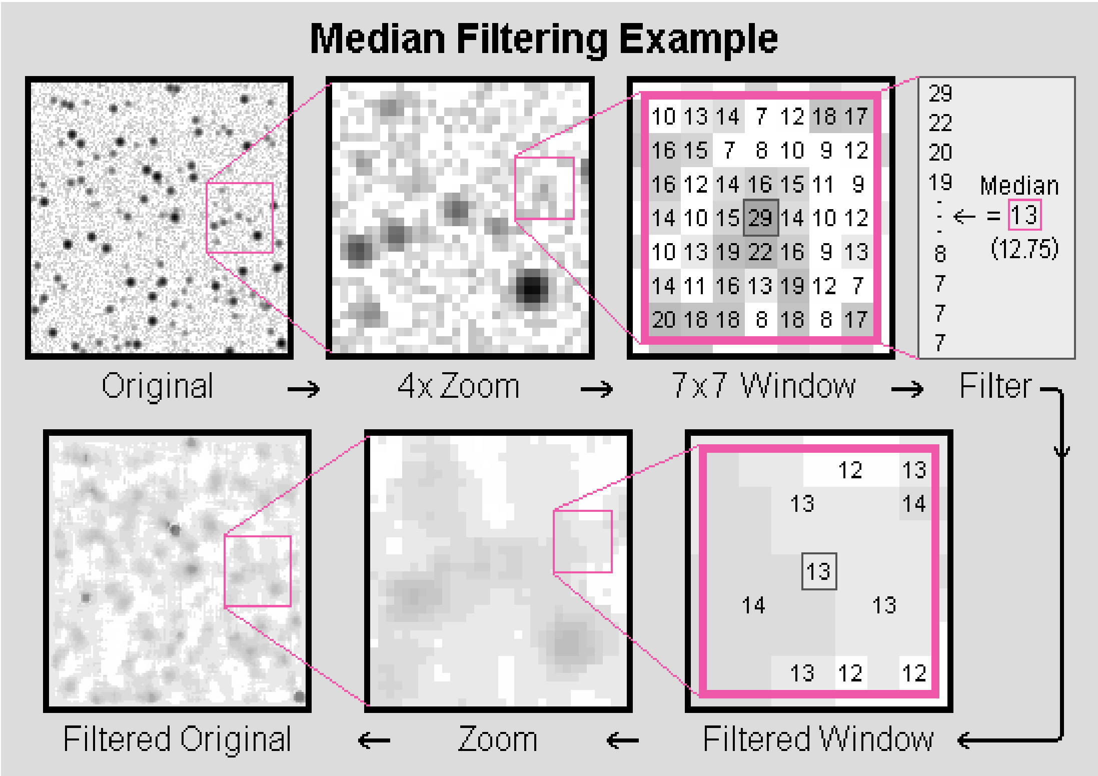
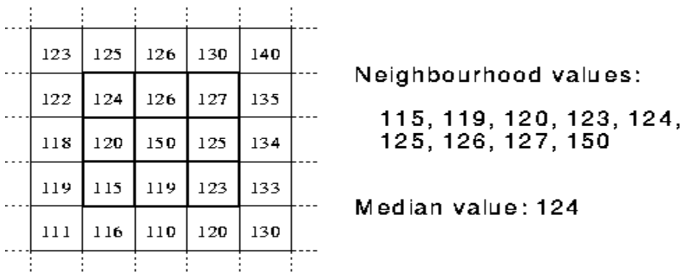



**Figure 2.3**: Median Filtering example

Like the mean filter, the median filter considers each pixel in the image in turn and looks at its nearby neighbors to decide whether or not it is representative of its surroundings. It replaces it with the *median* of those values. The median is calculated by first sorting all the pixel values from the surrounding neighborhood into numerical order and then replacing the pixel being considered with the middle pixel value. (If the neighborhood under consideration contains an even number of pixels, the average of the two middle pixel values is used).

**Figure 2.4**: Statistical data obtained



Calculating the median value of a pixel neighbourhood: As can be seen, the central pixel value of 150 is rather unrepresentative of the surrounding pixels and is replaced with the median value: 124. A 3×3 square neighbourhood is used here where larger neighbourhoods will produce more severe smoothing.

#### 2.3.2.1 NOISE

Noise is any undesirable signal. Noise is everywhere and thus we have to learn to live with it. Noise gets introduced into the data via any electrical system used for storage, transmission, and/or processing. In addition, nature will always plays a "noisy" trick or two with the data under observation. When encountering an image corrupted with noise you will want to improve its appearance for a specific application. The techniques applied are application-oriented. Also, the different procedures are related to the types of noise introduced to the image. Some examples of noise are: Gaussian or White, Rayleigh, Shot or Impulse, periodic, sinusoidal or coherent, uncorrelated, and granular.

#### 2.3.2.2 COMPARISON BETWEEN THE MEDIAN FILTER AND THE AVERAGE FILTER

Sometimes we are confused by median filter and average filter, thus let’s do some comparison between them. The median filter is a non-linear tool, while the average filter is a linear one. In smooth, uniform areas of the image, the median and the average will differ by very little. The median filter removes noise, while the average filter just spreads it around evenly. The performance of median filter is particularly better for removing impulse noise than average filter. As Figure 2.5, shown below are the original image and the same image after it has been corrupted by impulse noise at 10%. This means that 10% of its pixels were replaced by full white pixels. Also shown in Figure 2.7 are the median filtering results using 3x3 and 5x5 windows; three (3) iterations of 3x3 median filter applied to the noisy image; and finally for comparison, the result when applying an average filter as in Figure 2.8.

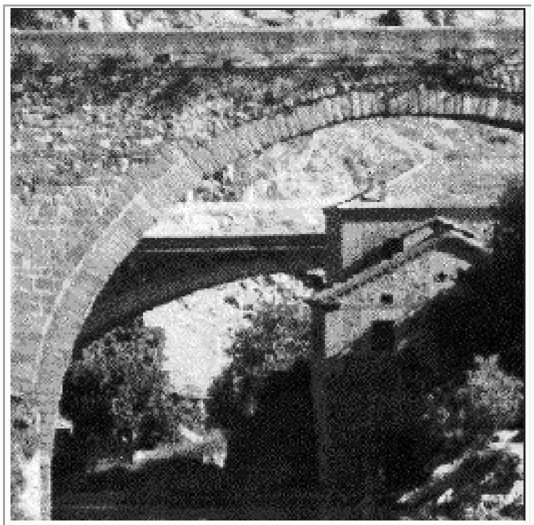


**Figure 2.5a:**Original image

**Figure 2.5b:**Corrupted by noise at 10 %

**Figure 2.6:**Average filter output

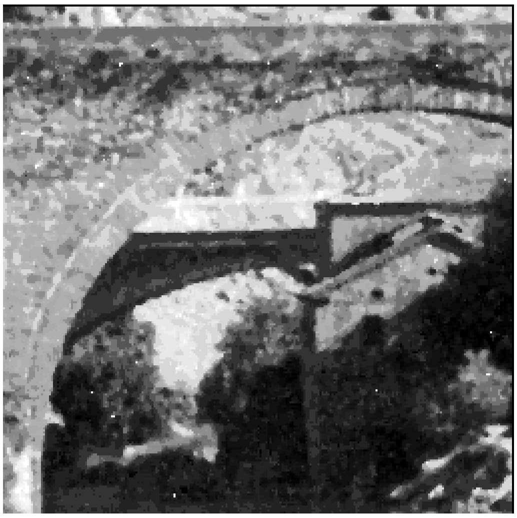
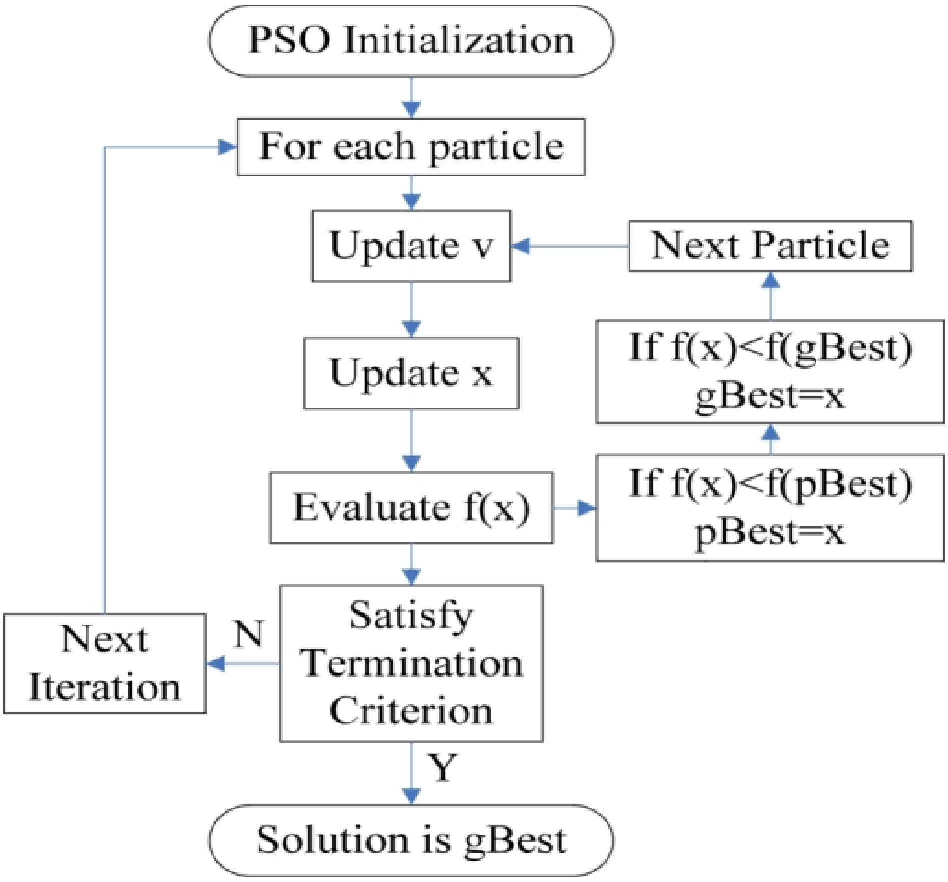


**Figure 2.7:**Median Filter output

# 3.PSO ALGORITHM

**PSO-** Particle Swarm Optimization

PSO is first initialized into a swarm of random particles. By following current optimal particle, all particles search in the solution space until the optimal solution is found. We define the notation adopted in this paper: assuming that the search space is $N$-dimensional, the number of particle is $n$, the $i$-th particle of the swarm is represented by the $N$-dimensional vector, $X_i = (x_{i1}, x_{i2}, \cdots, x_{iN})$ and the best particle of the swarm, i.e. the particle with the lowest function value, is denoted by index $g$. The best previous position (i.e. the position giving the best function value) of the $i$-th particle is recorded and represented by $X_i = (x_{i1}, x_{i2}, \cdots, x_{iN})$, and the position change (velocity) of the $i$-th particle is $V_i = (v_{i1}, v_{i2}, \cdots, v_{iN})$.A simple algorithm representing a PSO algorithm is given.



## 3.1 BASIC CONCEPTS

PSO is based on two fundamental disciplines: social science and computer science. In addition, PSO uses the swarm intel- ligence concept, which is the property of a system, whereby the collective behaviors of unsophisticated agents that are in- teracting locally with their environment create coherent global functional patterns. Therefore, the cornerstones of PSO can be described as follows

### 3.1.1 SOCIAL CONCEPTS

It is known that "*human intelligence results from social interaction*." Evaluation, comparison, and imitation of others, as well as learning from experience allow humans to adapt to the environment and determine op- timal patterns of behavior, attitudes, and suchlike. In addition, a second fundamental social concept indicates that "*culture and cognition are inseparable consequences of human sociality*." Culture is generated when individuals become more similar due to mutual social learning. The sweep of culture allows individ- uals to move towards more adaptive patterns of behavior.

## 3.2 SWARM INTELLIGENCE PRINCIPLES

Swarm Intelligence can be described by considering five fundamental principles.

1) *Proximity Principle*: the population should be able to carry out simple space and time computations.

2) *Quality Principle*: the population should be able to respond to quality factors in the environment.

3) *Diverse Response Principle*: the population should not commit its activity along excessively narrow channels.

4) *Stability Principle*: the population should not change its mode of behavior every time the environment changes

5) *Adaptability Principle*: the population should be able to change its behavior mode when it is worth the computa- tional price.

In PSO, the term "particles" refers to population members which are mass-less and volume-less (or with an arbitrarily small mass or volume) and are subject to velocities and accel- erations towards a better mode of behaviour.

## 3.3 COMPUTATIONAL CHARACTERISTICS

Swarm intelligence provides a useful paradigm for implementing adaptive systems. It is an extension of evolutionary computation and includes the softening parameterization of logical operators like AND, OR, and NOT. In particular, PSO is an extension, and a potentially important incarnation of *cellular automata* (CA). The particle swarm can be conceptualized as cells in CA, whose states change in many dimensions simultaneously. Both PSO and CA share the following computational attributes.

1) Individual particles (cells) are updated in parallel.

2) Each new value depends only on the previous value of the particle (cell) and its neighbors.

3) All updates are performed according to the same rules. Other algorithms also exist that are based on swarm intelli-

gence. The *ant colony optimization* (ACO) algorithm was in- troduced by Dorigo in 1992. It is a probabilistic technique for solving computational problems, which can be reduced to finding good paths through graphs. It is inspired by the behavior of ants in finding paths from the colony to the food. In the real world, ants initially wander randomly, and upon finding food, they return to their colony while laying down pheromone trails. If other ants find such a path, they are likely not to keep traveling at random, but rather follow the trail, returning and reinforcing it if they eventually find food . However, the pheromone trail starts to evaporate over time, therefore reducing its attrac- tive strength. The more time it takes for an ant to travel down the path and back again, the longer it takes for the pheromones to evaporate. A short path, by comparison, gets marched over faster, and thus the pheromone density remains high as it is laid on the path as fast as it can evaporate. Pheromone evaporation also has the advantage of avoiding the convergence to a locally optimal solution. If there were no evaporation at all, the paths chosen by the first ants would tend to be excessively attractive to the following ones. In that case, the exploration of the so- lution space would be constrained. Thus, when one ant finds a short path from the colony to a food source (i.e., a good solu- tion), other ants are more likely to follow that path, and positive feedback eventually leaves all the ants following a single path. The idea of the ant colony algorithm is to mimic this behavior with “simulated ants” walking around the graph representing the problem to solve. ACO algorithms have an advantage over simulated annealing (SA) and GA approaches when the graph may change dynamically, since the ant colony algorithm can run continuously and adapt to changes in real time. *Stochastic diffusion search* (SDS) is another method from the family of swarm intelligence, which was first introduced by Bishop in 1989 as a population-based, pattern-matching algorithm. The agents perform cheap, partial evaluations of a hypothesis (a candidate solution to the search problem). They then share information about hypotheses (diffusion of information) through direct one-to-one communication. As a result of the diffusion mechanism, high-quality solutions can be identified from clusters of agents with the same hypothesis. In addition to the above techniques, efforts have been made in the past few years to develop new models for swarm intelligence systems, such as a *honey bee colony* and *bacteria foraging*. The honey bee colony is considered as an intelligent system that is composed of a large number of sim- plified units (particles). Working together, the particles give the system some intelligent behavior. Recently, research has been conducted on using the honey bee model to solve optimization problems. This can be viewed as modeling the bee foraging, in which the amount of honey has to be maximized within a min- imal time and smaller number of scouts. Bacteria foraging emulates the social foraging behavior of bacteria by models that are based on the foraging principles theory. In this case, foraging is considered as an optimiza- tion process in which a bacterium (particle) seeks to maximize the collected energy per unit foraging time. Bacteria foraging provides a link between the evolutionary computation in a social foraging environment and the distributed nongradient optimization algorithms that could be useful for global optimization over noisy conditions. This algorithm has been recently applied to power systems as well as adaptive control applications. This approach enables to determinate the "optimum" number of the thresholds and simultaneously the optimal thresholds of three criteria: the between-class variances criterion, the minimum error criterion and the entropy criterion

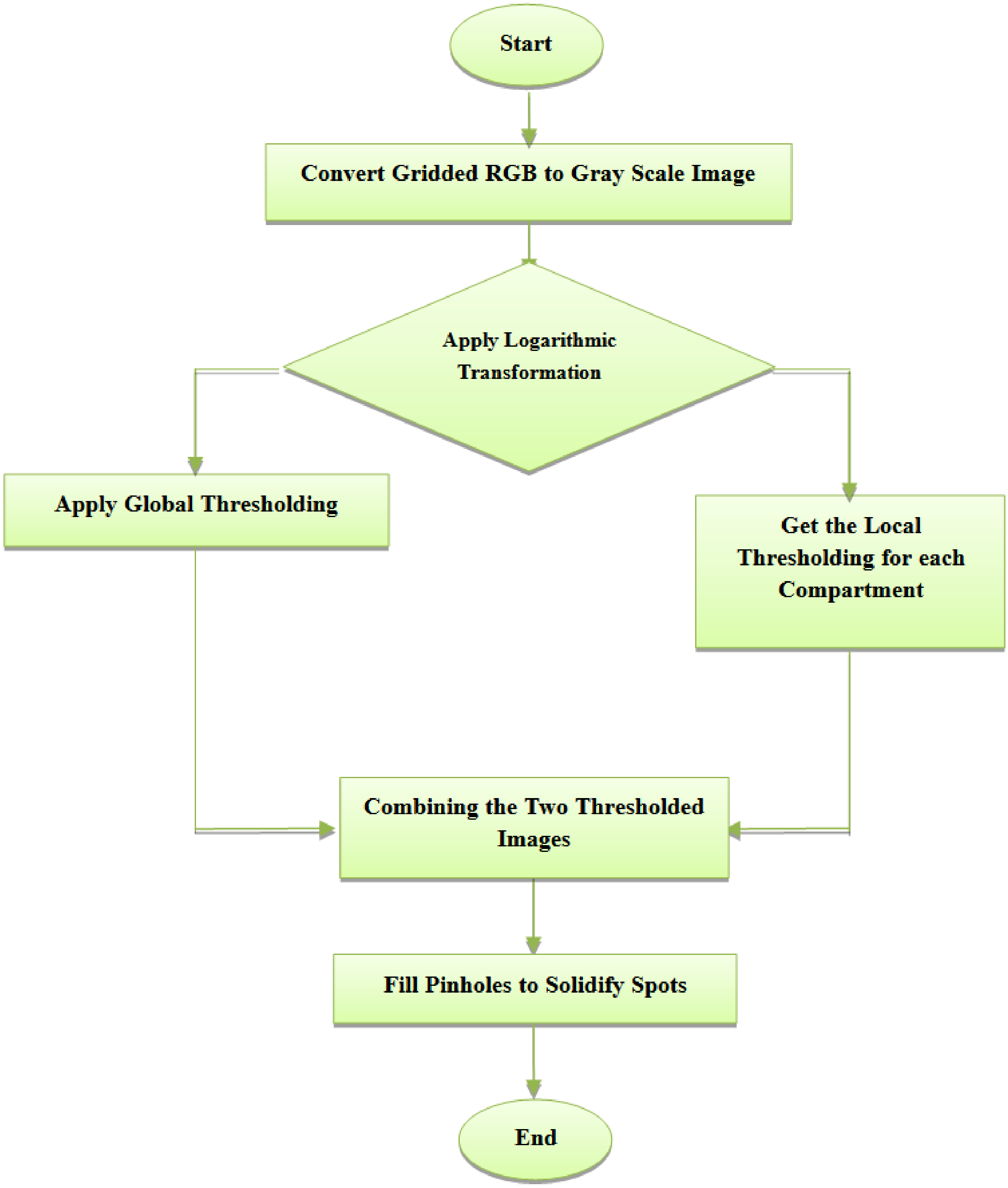



Figure 3.1 Grey Threshold Image segmentation

# 4.ROBOT INITIALIZATION

Once the object has been identified, a robot initialized. Then the weight of the object is calculated in terms of its area i.e, the object that has a larger area than the other object is assumed to be of higher weight and the latter as a smaller weight.According to the weight of the object, the required number of robots are then initialized subsequently.

## 4.1 CALCULATION OF WEIGHT

The area of the image(here image=object) is calculated using the area function and is assumed to be the equivalent for weight of the object.

## 4.2 INITIALIZATION OF ROBOTS

Irrespective of the weight calculated, once an object is detected, one robot is initialized. Then, according to the weight, the required number of robots are initialized. If the weight of the object detected exceeds the capacity of the first robot, then the second robot is initialized. Even then if the weight is too large for the combined capacity of the two robots, the third robot is initialized and so on for subsequent robots. On the other hand, if the object is not in the database and is not detected, none of the robots are initialized. For adding new images to the database, a separate program is written for enrollment.

# 5.SIMULATION TOOL(MATLAB)

MATLAB is an interactive software package which was developed to perform numerical calculations on vectors and matrices. Initially, it was simply a MATrix LABoratory. However, today it is much more powerful.It has the following features(i)It can do quite sophisticated graphics in two and three dimensions.(ii)It contains a high level programming language (a “baby C”) which makes it quite easy to code complicated algorithms involving vectors and matrices(iii)It can numerically solve nonlinear initial value ordinary differential equations(iv)It can numerically solve nonlinear boundary value ordinary differential equations( v)It contains a wide variety of toolboxes which allow it to perform a wide range of applications from science and engineering. Since users can write their own toolboxes, the breadth of applications is quite amazing.Mathematics is the basic building block of science and engineering, and MATLAB makes it easy to handle many of the computations involved. And this access is available by using only a small number of commands and function because MATLAB's basic data element is a matrix (or an array).

## 5.1 IMAGE TYPES

The toolbox supports four types of images:

- Gray-scale images
- Binary images
- Indexed images
- RGB images

Most monochrome image processing operations are carried out using binary or gray-scale images, so our initial focus is on these two image types.

### 5.1.1 GRAY-SCALE IMAGES

A gray-scale image is a data matrix whose values represent shades of gray. When the elements of a gray-scale image are of class uint8 or uint16, they have integer values in the range [0, 255] or [0, 65535], respectively. If the image is of class double or single, the values are floating-point numbers. Values of double and single gray-scale images normally are scaled in the range [0, 1], although other ranges can be used.

### 5.1.2 BINARY IMAGES

Binary images have a very specific meaning in MATLAB. A binary image is a logical array of 0s and 1s. Thus, an array of 0s and 1s whose values are of data class, say, uint8, is not considered a binary image in MATLAB. A numeric array is converted to binary using function logical. Thus, if A is a numeric array consisting of 0s and 1s, we create a logical array B using the statement

B = logical(A)

If A contains elements other than 0s and 1s, the logical function converts all nonzero quantities to logical 1s and all entries with value 0 to logical 0s.

## 5.2 IMAGE PROCESSING IN MATLAB

In MATLAB a binary and gray-scale image is represented by one 2-dimensional array, whereas a color image is represented by a 3-dimensional array (one 2-dimensional array for each of the color planes or color channels red, green and blue): The origin of the image is in the upper left and the size of the image is defined by the parameter width (number of columns of the array) and height (number of rows of the array).

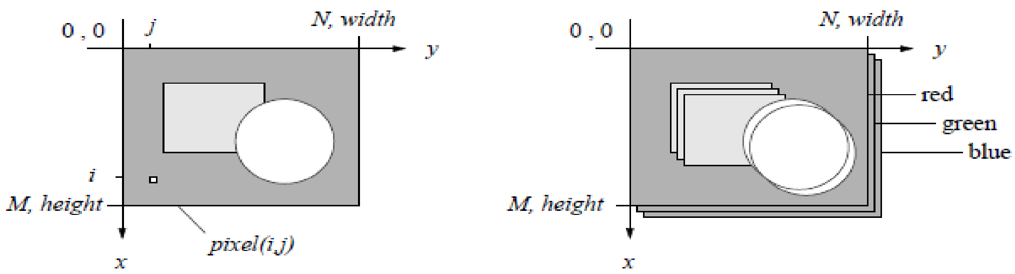



**Figure 5.1:**Image representation

In the case of uint8 images, the logical flag must be turned on, to be recognized as binary image (for details see below). Many of the predefined functions, e.g. imadd(img1, img2) which adds two images, just truncates data values to 255 on uint8 arrays make sure if that is what you want.To avoid problems with data types, especially when working with predefined image processing functions, it is advisory to work with type double and make sure that data is scaled to the range 0 ... 1.0.

### 5.3 ADVANTAGES OF MATLAB

1.Ease of use

2.Platform independence

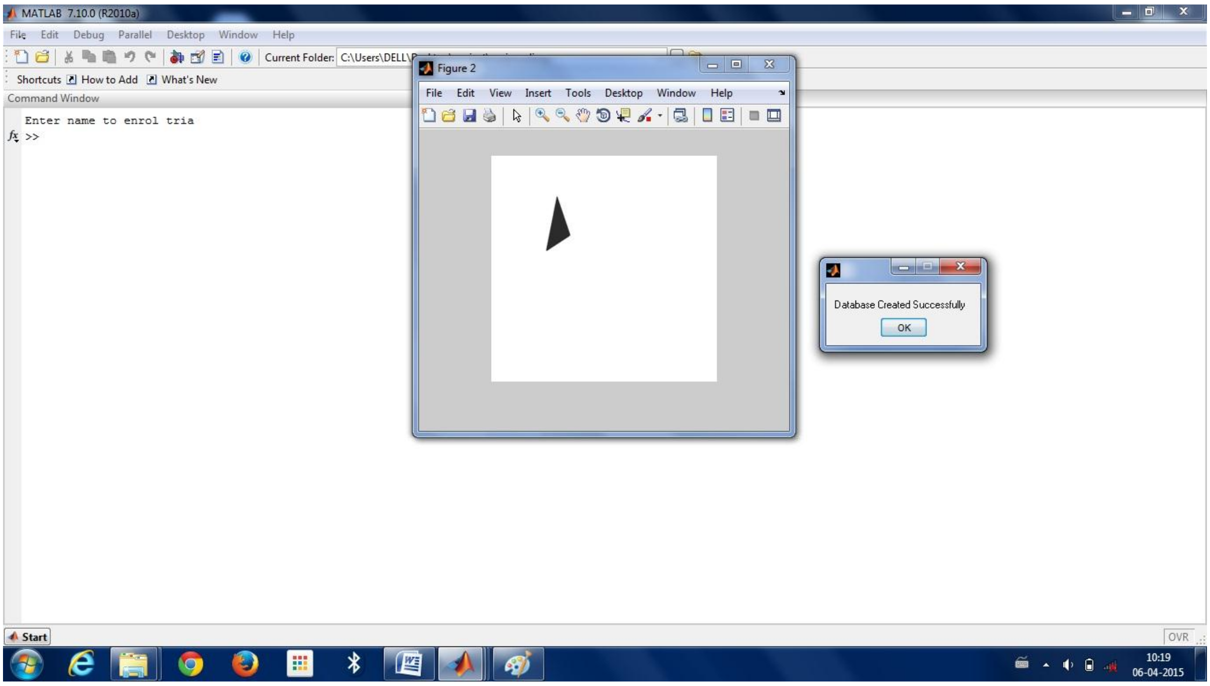
3.Predefined functions

4.Plotting

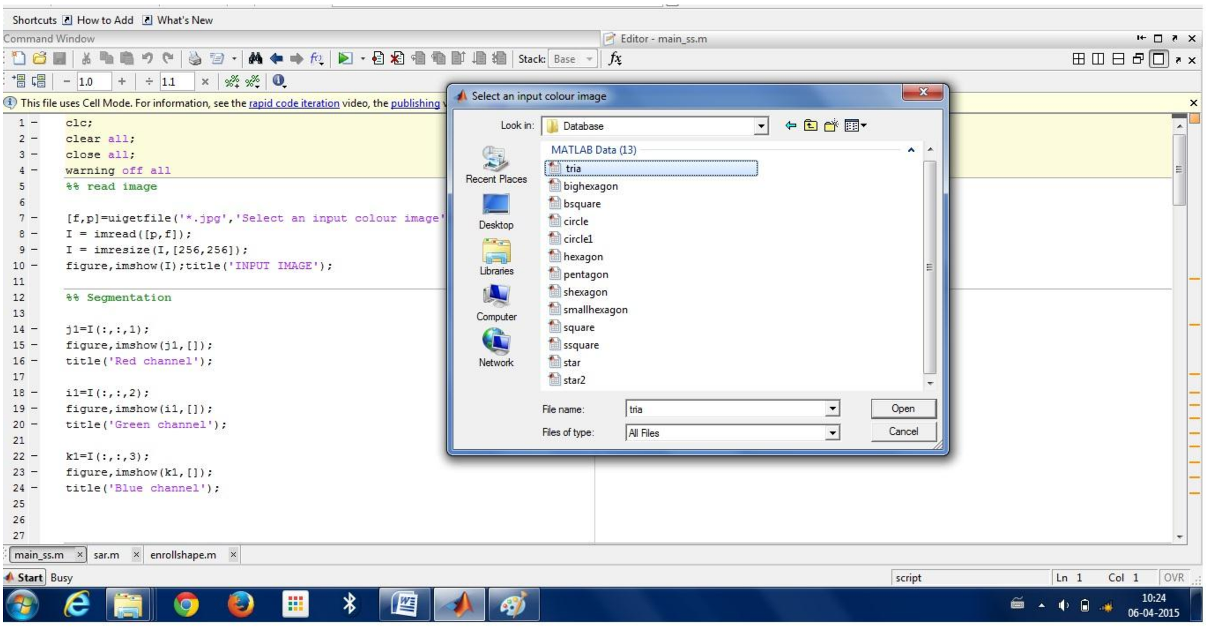
# 6. SIMULATED RESULTS

## IMAGE TO BE ADDED TO THE DATABASE

## IMAGE ENROLLMENT



# IMAGE ADDED TO THE DATABASE



# INPUT IMAGE

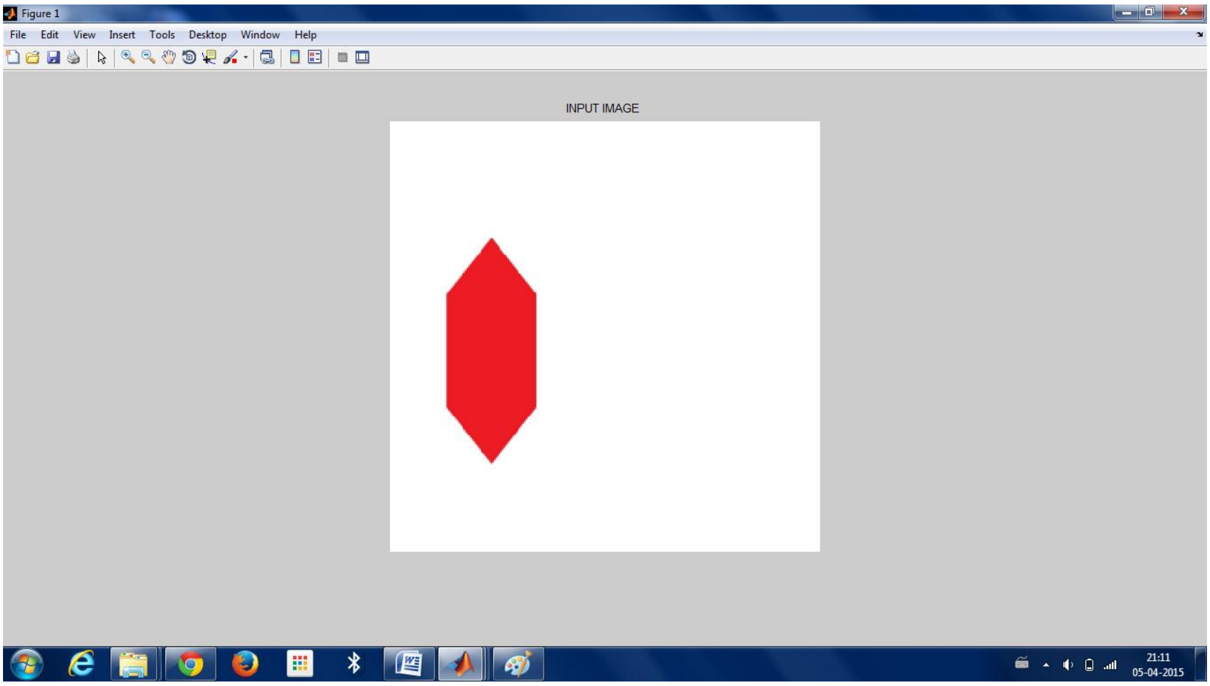
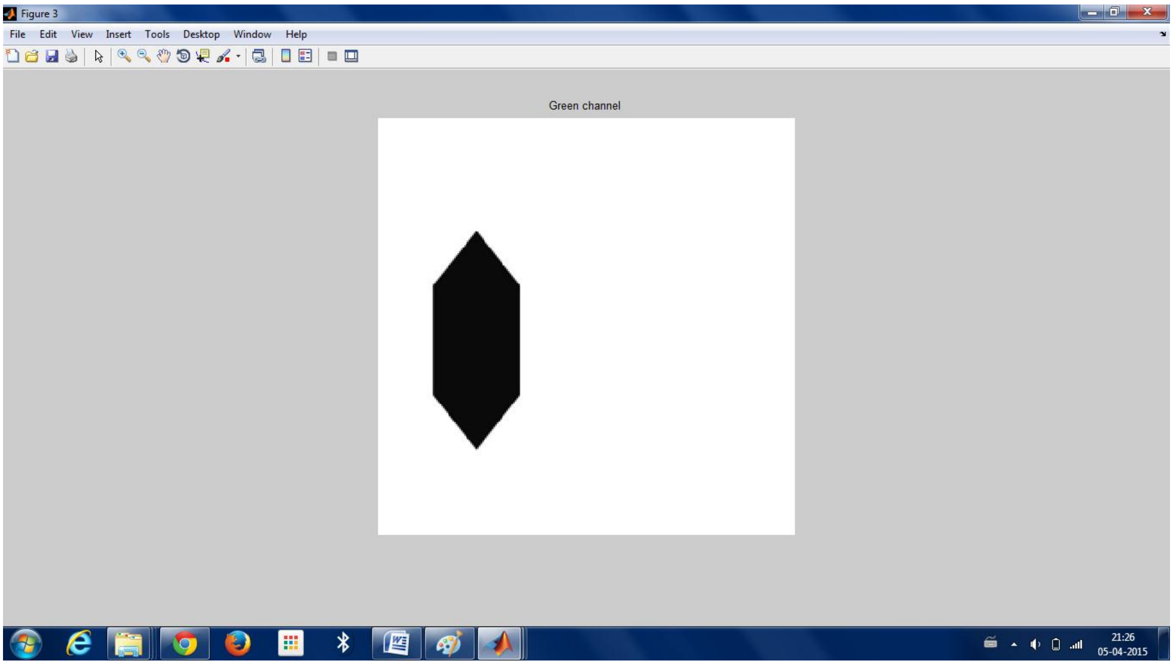
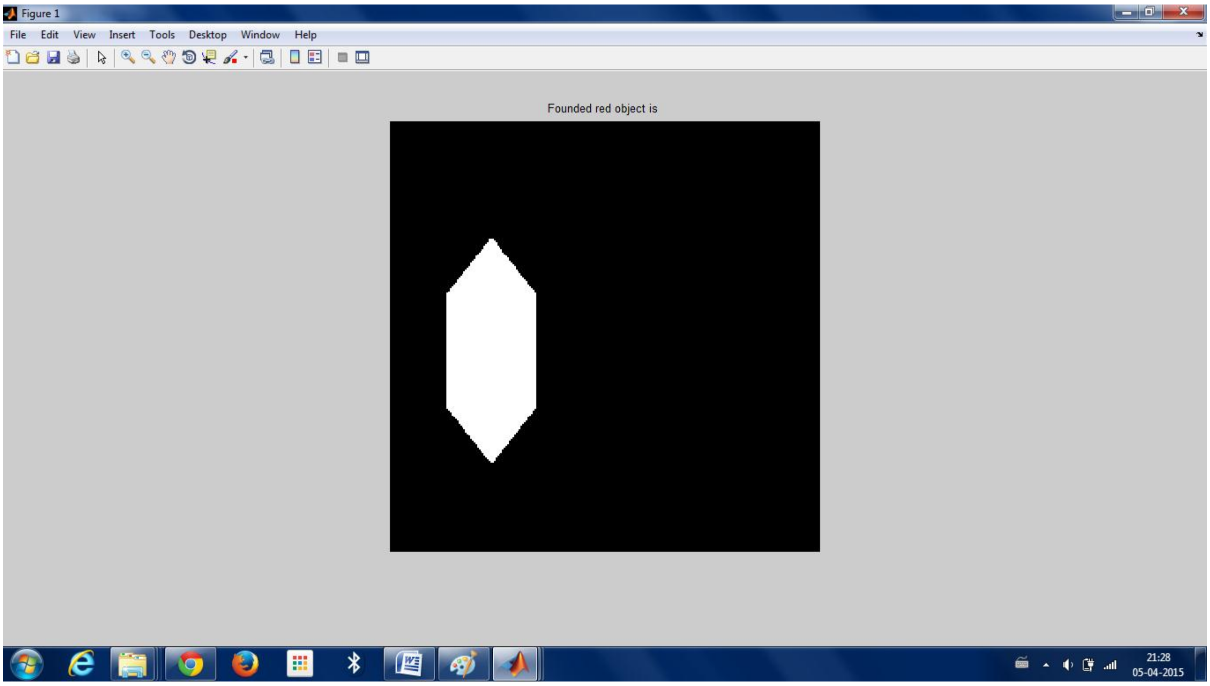


Figure 1
File Edit View Insert Tools Desktop Window Help
INPUT IMAGE
21:11
05-04-2015

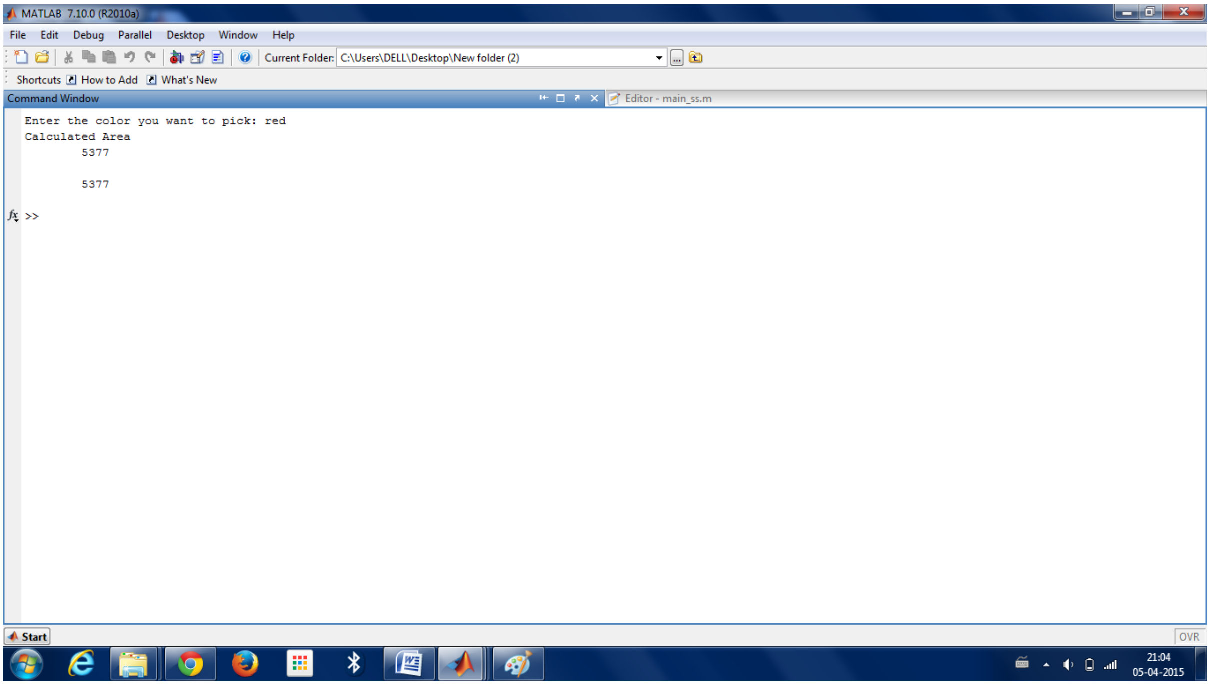
# SEGMENTATION IN DIFFERENT BACKGROUNDS



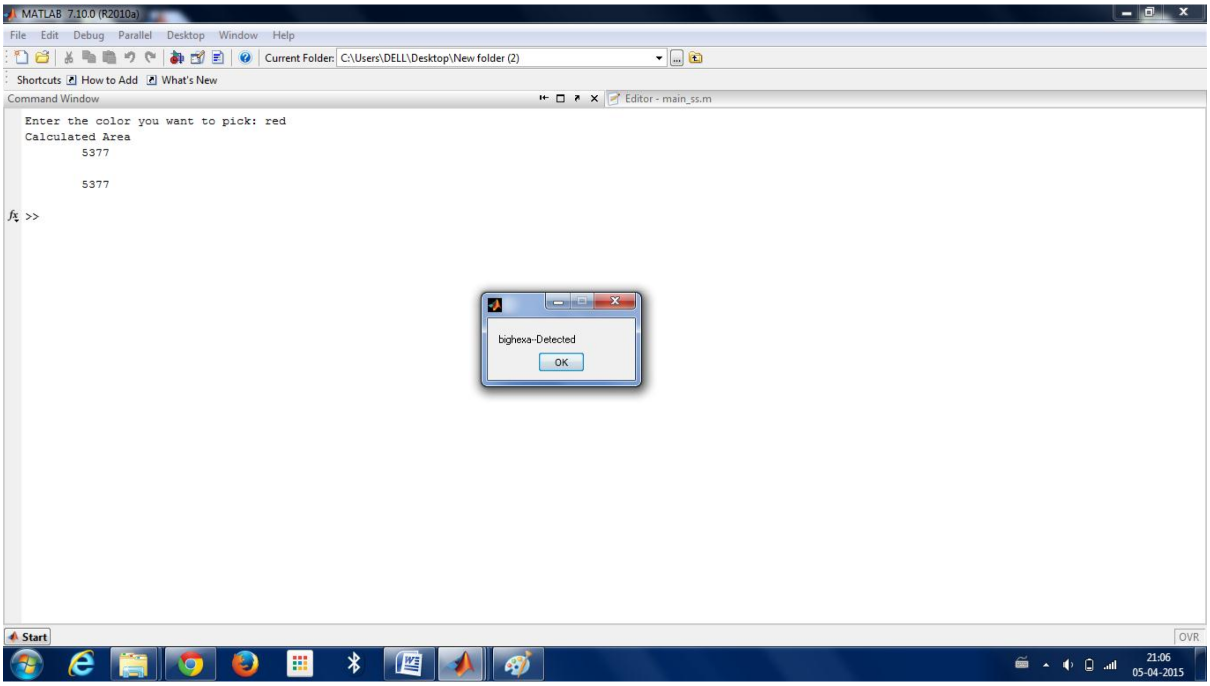
# OBJECT IDENTIFIED



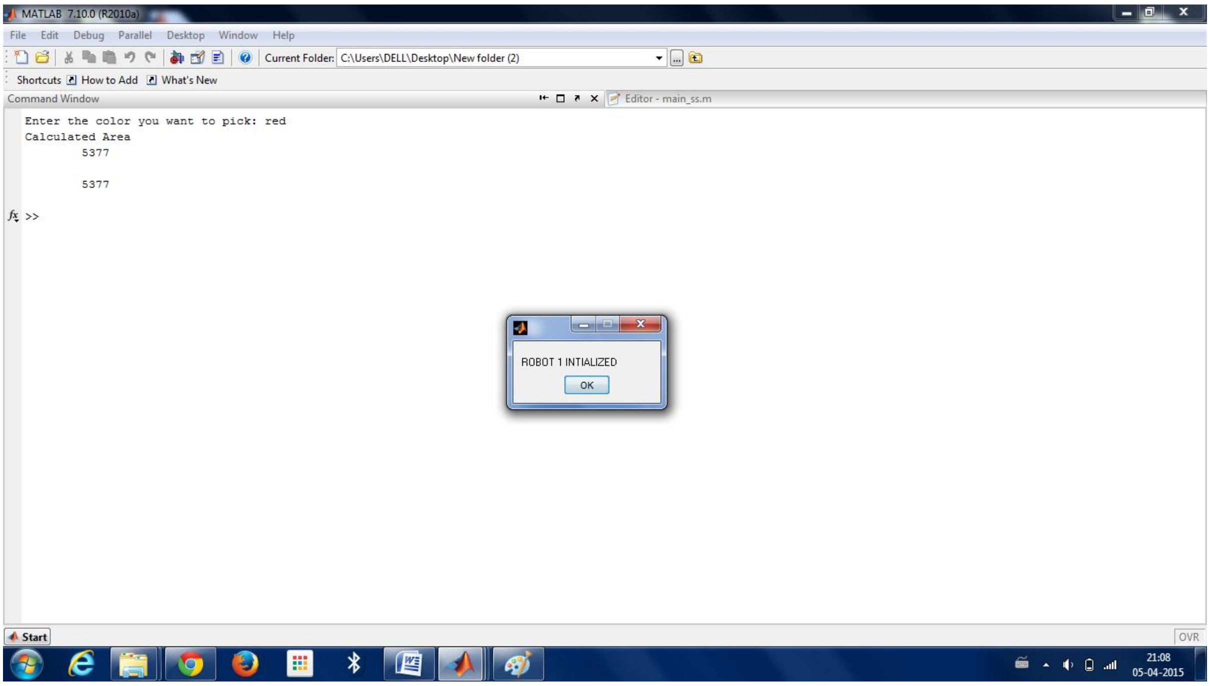
# COLOR SELECTION



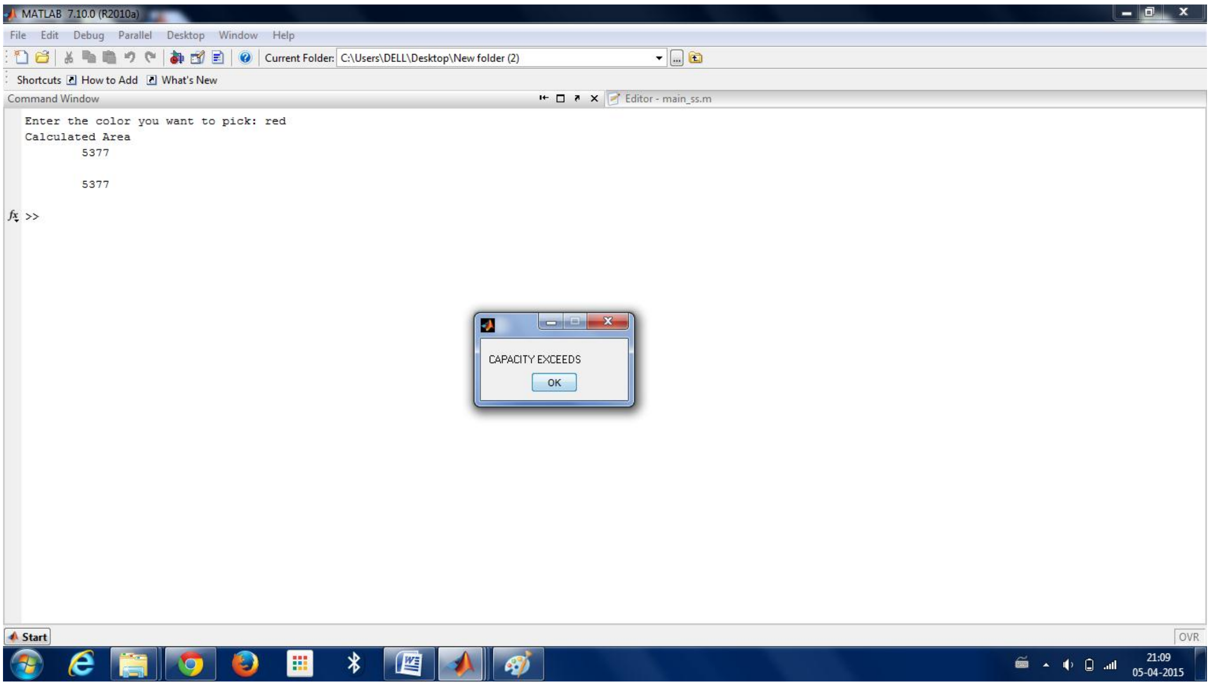
# AREA CALCULATION AND DETECTION



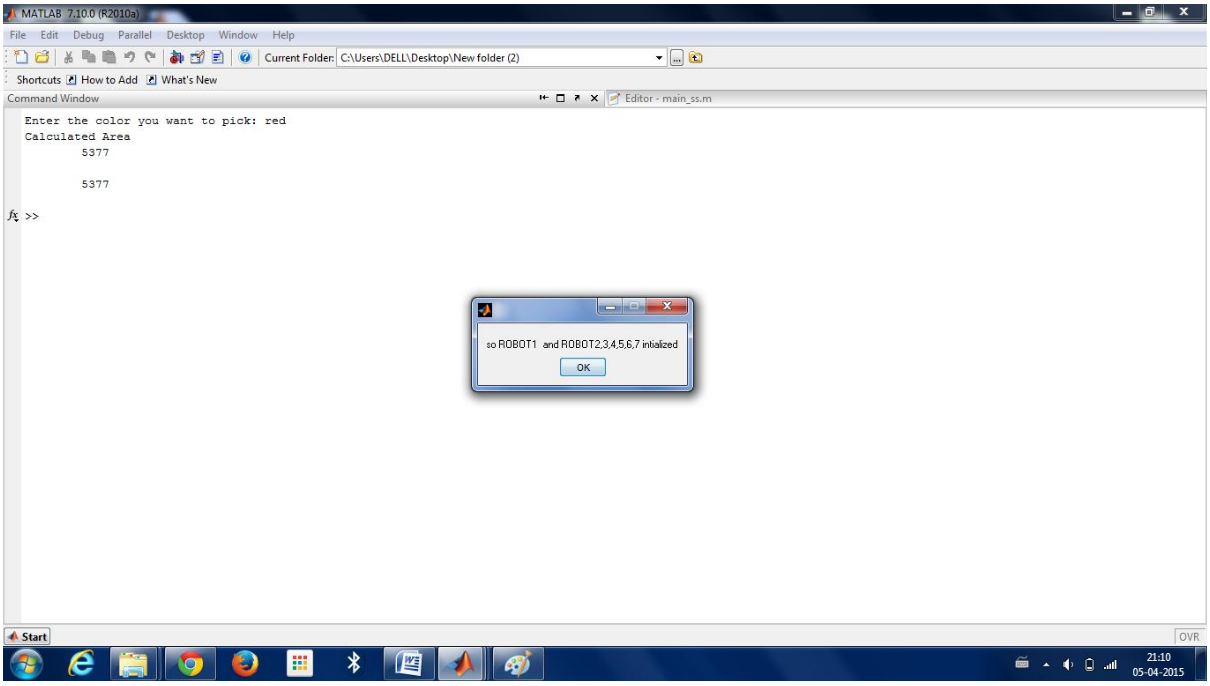
# INITIALIZATION OF FIRST ROBOT



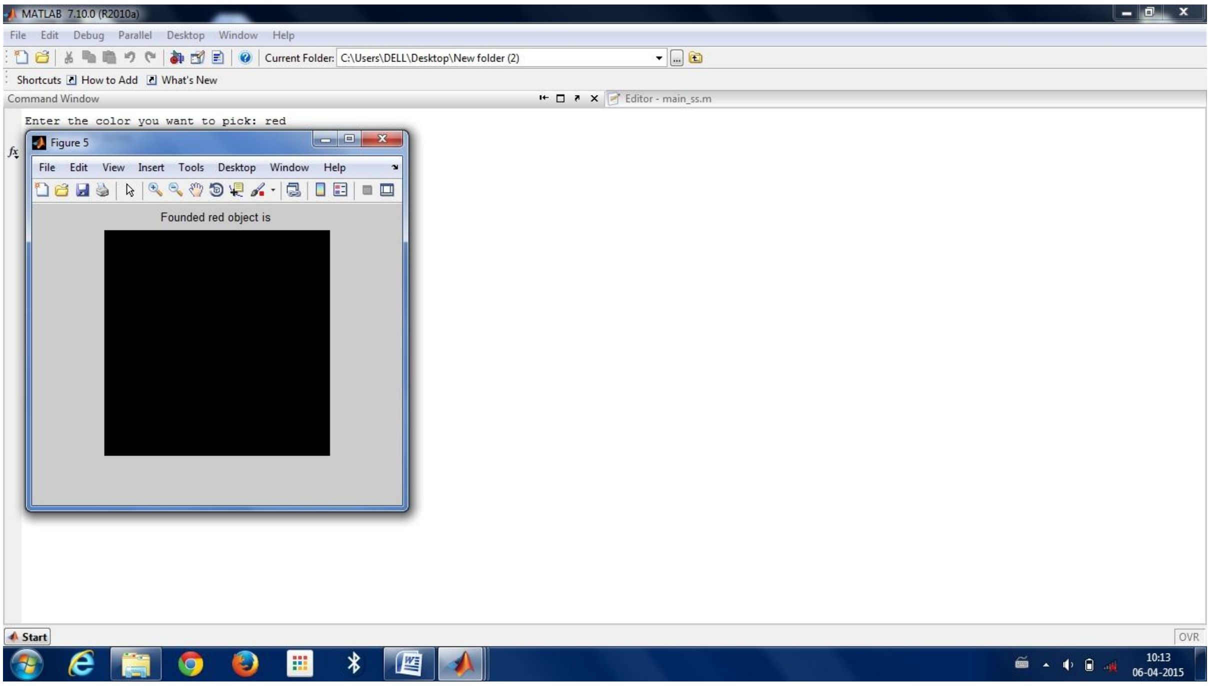
# DIALOG BOX WHEN CAPACITY EXCEEDS



# INITIALIZATION OD ADDITIONAL ROBOTS

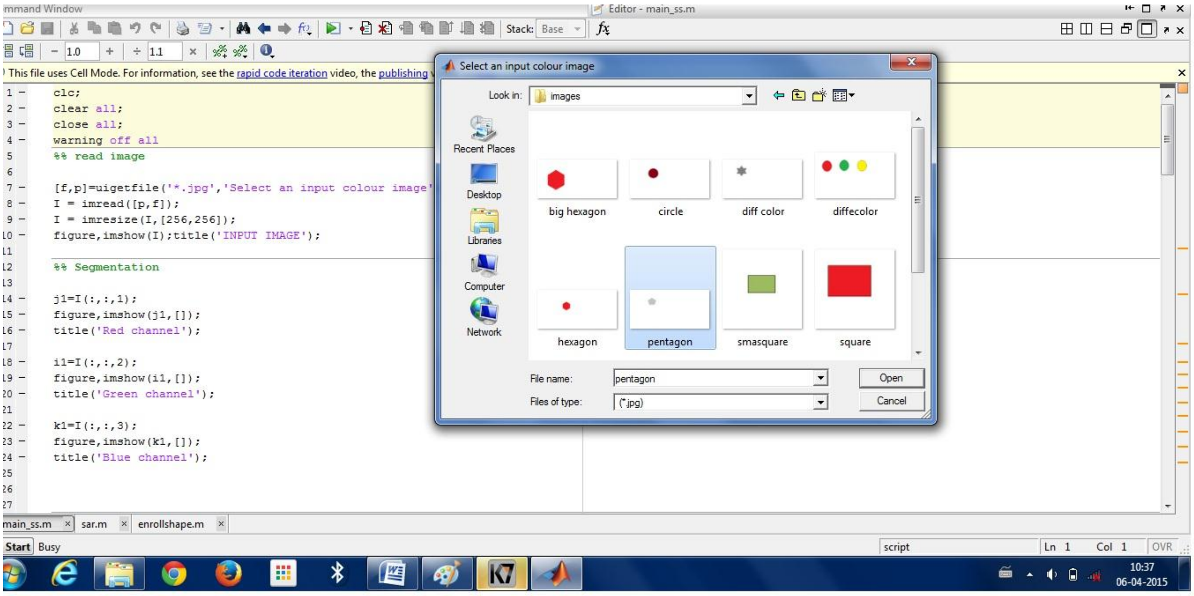


**IN CASE THE IMAGE IS NOT IDENTIFIED**
(If color mismatches or image not in database)



**OBJECT ENTERED: GREY PENTAGON**

**COLOUR ENTERED:RED**



## 7. FUTURE SCOPE

Swarms of robots acting together to carry out jobs could provide new opportunities for humans to harness the power of machines, the ability to control robot swarms could prove hugely beneficial in a range of contexts, from military to medical. There are some researches that have demonstrated that the swarm can carry out simple fetching and carrying tasks, by grouping around an object and working together to push it across a surface. The robots can also group themselves together into a single cluster after being scattered across a room, and organize themselves by order of priority. Swarming robots could have important roles to play in the future of micromedicine, as 'nanobots' are developed for non-invasive treatment of humans. On a larger scale, they could play a part in military, or search and rescue operations, acting together in areas where it would be too dangerous or impractical for humans to go. In industry too, robot swarms could be put to use, improving manufacturing processes and workplace safety. For example, if the robots are being asked to group together, each robot only needs to be able to work out if there is another robot in front of it. If there is, it turns on the spot; if there isn't, it moves in a wider circle until it finds one. The key is to work out what is the minimum amount of information needed by the robot to accomplish its task. That's important because it means the robot may not need any memory, and possibly not even a processing unit, so this technology could work for nanoscale robots, for example in medical applications. Future developments of this work are related further improve our mathematical model by adding more Quality of Service parameters. In the future, there will be improvement in our algorithms to reallocate resources according to requirement or to the existing status of the Cloud in order to optimize resource usage.

# 8. CONCLUSION

Swarm robotics has several possible applications, including  exploration, surveillance, search and rescue, humanitarian demining, intrusion tracking, cleaning, inspection and transportation of large objects. Despite their potential to be robust, scalable and flexible, up to now, swarm robotics systems have never been used tackle a real-world application and are still confined to the world of academic research. At the current state of development of the swarm robotics field, the focus is mostly on obtaining desired collective behaviors and understanding their properties. Foraging is the most used testbed application for swarm robotics systems. Robots have to retrieve “prey” objects from an environment and bring them back to a “nest”. Foraging can be considered as an abstraction with many point in common with more complex applications, such as demining and search and rescue. Foraging is also used to investigate the effect of interference in swarm robotics systems . In particular, foraging is commonly used as a testbed for collective exploration ,collective transport and collective decision-making . We foresee that, as swarm robotics is further developed and as it is used to tackle real-world applications, the need for a swarm engineering will increase. The goal of this project was not only to present the significant works on methods and collective behaviors in swarm robotics, but also to propose a systematic categorization of their aspects. We think that such categorization effort is a necessary step for the development of a swarm engineering.

# 9. REFERENCES


1.www.swarmbot.org

2. http://www.scholarpedia.org/article/Swarm_robotics

3. "A self-organizing thousand-robot swarm".*Harvard*. 14 August 2014. Retrieved 16 August2014.

4.Arvin, F.; Murray, J.C.; Licheng Shi; Chun Zhang; Shigang Yue, "Development of an autonomous micro robot for swarm robotics," Mechatronics and Automation (ICMA), 2014 IEEE International Conference.

5.Kennedy, J.; Eberhart, R. (1995). "Particle Swarm Optimization". *Proceedings of IEEE International Conference on Neural Networks* .

6.Kennedy, J. (1997). "The particle swarm: social adaptation of knowledge". *Proceedings of IEEE International Conference on Evolutionary Computation*.

7.Pal, N.R. and Pal, S.K. 1993. A review on image segmentation techniques. Pattern Recognition, 26 (9), 1277 – 1294.

8.G. Beni. From swarm intelligence to swarm robotics. In *Swarm Robotics*, LNCS 3342, pp. 1–9, 2005. Springer.

9. http://in.mathworks.com/products/image/code-examples.html

10.Azriel Rosenfeld, *Picture Processing by Computer*, New York: Academic Press, 1969

11. http://robotics.eecs.berkeley.edu/~sastry/ee20/gacode.html

12. http://stackoverflow.com/questions/13193248/how-to-make-a-gaussian-filter-in-matlab

13. http://www.codemiles.com/matlab-examples/median-filter-to-image-t7273.html

14. M. Reyes-Sierra, C.A.C. Coello, “Multi-objective Particle Swarm Optimizers: A survey of the State-of-the-Art.”, Int. Journal of Comput. Intelligence Research, Vol. 2