

# **Comparison of the Edge Detection Methods to Detect, Identify and Locate the Obstacles for Agricultural Robotic Vehicles**

**Anas Qasim Mahdi**

Submitted to the  
Institute of Graduate Studies and Research  
in partial fulfillment of the requirements for the Degree of

Master of Science  
in  
Computer Engineering

Eastern Mediterranean University  
January 2014  
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

---

Prof. Dr. Elvan Yılmaz  
Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Science in Computer Engineering.

---

Prof. Dr. Işık Aybay  
Chair, Department of Computer Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Computer Engineering.

---

Asst. Prof. Dr. Mehmet Bodur  
Supervisor

---

Examining Committee

1. Asst. Prof. Dr. Adnan Acan

2. Asst. Prof. Dr. Mehmet Bodur

3. Asst. Prof. Dr. Ahmet Ünveren

## ABSTRACT

The obstacle detection in an agricultural field is an important step of the automation of the plantation. There are already developed autonomous agricultural vehicles that can track a path, and perform the specified processes on the plantation fields. These autonomous agricultural robotic machines need an upper level of control, which is mostly performed manually, for the design of the reference paths. Detection of the agricultural obstacles is necessary to accomplish these manual tasks in an automatic manner. In this study, statistical methods are employed to determine which of the five well-known edge-detection methods is best, for the high-level path planning in an agricultural automation of autonomous agricultural vehicles depending on field and image properties.

**Keywords:** agricultural robotic, edge detection techniques, Canny, Prewitt, Robert, Sobel, obstacle detection.

## ÖZ

Tarım alanlarında engel tesbiti tarımsal otomasyonu önemli bir basamağıdır. Tarım arazilerinde verilen bir yolu takip ederek belirtilen işlemleri uygulamak üzere geliştirilmiş tarım araçları şimdiden mevcuttur. Bu otonom tarım robotları şimdilik elle gerçekleştirilen üst düzeyde bir yol tasarımına gerek duyarlar. İşin tümüyle otomasyonu için tarımsal engellerin tesbitini otomatik olarak yapabilmek gerekir. Bu çalışmada, tarımsal üst düzey yol planlaması açısından en iyi kenar belirleme yöntemi araştırılmış, istatistiksel yöntemler ile yaygın bilinen beş yöntemin arazi ve resim özelliklerine bağlı olarak hangisinin iyi sonuç verdiği belirlenmiştir.

I dedicate this thesis to my lovely mother, my Great father, my dear aunt, best sisters, brothers, and to all my friends.

## **ACKNOWLEDGMENT**

In the name of God I and all trust in God. First of all, great thanks to my supervisor Professor Mehmet Bodur for his great help, supervision. I wish to dedicate this thesis with love to my parents for giving me the freedom to reach my path and to pursue my interests; and also thanks to my wonderful sisters Noor, and Enas and my brothers Ahmed and, Mohamed amen for giving me love and confidence, believing that I am always successful.

Special thanks to my friends Humam M. Jassim, Haydar M. Jassim, Mohammed Namik, Sinan Hazim, Ahmed Hanny, Omer Heyman, Liwaa Hussein, Ahmed Mahmoud, Ali al Faisal, Mustafa Ibraheem, Ghassan Qas Marrogy and to all my friends for their help and support.

# TABLE OF CONTENTS

ABSTRACT.....	iii
ÖZ.....	iv
DEDICATION.....	v
ACKNOWLEDGMENT.....	vi
LIST OF FIGURES.....	x
1 INTRODUCTION.....	1
1.1 Contents Of Sections.....	3
2 AGRICULTURE ROBOT AND IMAGE PROCESSING.....	4
3 EDGE DETECTION.....	7
3.1 Robert Edge Detection Technique.....	8
3.2 Sobel Edge Detection Techniques.....	9
3.3 Prewitt Edge Detection Techniques.....	11
3.4 Canny Edge Detection Techniques.....	12
4 EDGE DETECTION TECHNIQUES ON THE FIELD IMAGES.....	17
4.1 Obtaining Satellite Image And Edge Detection.....	17
4.2 Image Processes On The Field Image Extraction.....	18
4.4 Image Processing To Detect Obstacle(s).....	19
4.5 Error Evaluation.....	22
4.6 Results And Discussion.....	22
4.7 Determine The Location (Coordinates) And Size Of The Obstacles.....	35

5 CONCLUSION .....	38
REFERENCES.....	40



## LIST OF TABLE

Table 4.1 Data regarding correct obstacle detection section.....	.32
Table 4.2 Data regarding missed and wrong obstacle detection section .....	34
Table 4.3 Data regarding obstacle coordinates and radius .....	37

## LIST OF FIGURES

Figure 1: Agriculture Robot while recognizing the bad grass and weeds in the fields.....	5
Figure 2: Semi-circle into 5 regions.....	15
Figure 3: Sample API key is used to obtain Satellite images from Google maps. ....	17
Figure 4: a) Original image b) Binary image,.....	18
Figure 5: a) Binary image with undesired segments, b) undesired segment are removed from outside, c) desired field free off undesired objects, d) inverted image ....	19
Figure 6: Block diagram showing the Matlab steps processing for edge detection techniques.....	21
Figure 7 : Ten Test Fields with Different Size, Shape and Complexity .....	22
Figure 8: Canny edge detection technique .....	23
Figure 9: Canny edge detection technique for all samples false positive errors 20% .....	24
Figure10:Canny edge detection technique for all samples false negative errors 20%.....	24
Figure 6: Prewitt edge detection technique.....	25
Figure 7: Prewitt edge detection technique for all samples false positive errors 17% ....	26
Figure 8: Prewitt edge detection technique for all samples false negative errors 9% .....	26
Figure 9: Robert edge detection technique .....	27
Figure 10: Robert edge detection technique for all samples false positive errors 35% ...	28
Figure 11: Robert edge detection technique for all samples false negative errors 6.5% .	28
Figure 17: Sobel edge detection technique .....	29
Figure 18: Sobel edge detection technique for all samples false positive errors 16% .....	30
Figure 19: Sobel edge detection technique for all samples false negative errors 6.5% ...	30
Figure 20: All techniques for all of the samples false negative errors and false positive errors.....	31

Figure 12: Mapping of field coordinates to the center pixel of obstacles in original  
image.....36

# Chapter 1

## INTRODUCTION

Robotics has been considered as an important tool in supporting the production, development and quality insurance in every fields of industry such as military robots, healthcare robots and entertainment robots that perform most of the human activities [1]. In 1980's the production of robots became limited due to some reasons.

Currently with the rapid developments in science and technology today's robots possess capabilities and skills of high-end applications. These robots can be developed either to assist a farmer in cleansing the soil or for elimination of pests etc. [2]. The agricultural industry started to use autonomous agricultural vehicles as agricultural robots, which can track a predefined desired path while applying an agricultural process to the plantation field. They actually provide higher accuracy, lower cost and improved performance, at a higher speed than human.

In order to satisfy market's demands to improve productivity and quality more advanced agricultural production methods are required. Efficiency in plowing, seeding, harvesting etc., are quite essential, and to reach this goal there is a rapid trend to apply autonomous agricultural machineries in these crucial agricultural tasks. In parallel to these requests of agricultural industry, many algorithms and approaches has been produced to detect and annotate obstacles and objects in agricultural fields from satellite images [3]. To reduce the ambiguity in pictures and

in global features of the image, learning discriminative image patches has been provided for the recognition of given object classes. These classes use discriminative training of log-linear models to image patch histograms [4], to identify background and object regions. Path planning for an automated agricultural plantation system requires several levels of subsystems, such as recognition and planning of agricultural areas, detection of obstacles, and determination of optimum paths for a number of agricultural applications on soil and plants, path tracking, maneuver in the path terminations, etc. Subsystems that are implemented as an application layer needs considerable information exchange for a successful and efficient operation of the overall system.

Many of these subsystems have been covered in the literature from advanced coverage path planning algorithms to optimization and robotic steering for autonomous agricultural vehicles [3]. As far as we have searched, there are methods in the literature that proposed some solutions for the automatic steering, however none of them has provided precise globally positioning with annotation methods for a satellite image of an agricultural field. Many researchers considered the obstacles to be processed manually, which results in reduced information-exchange possibilities between the subsystems. In this study, comparison between four edge detection algorithms, namely Canny, Prewitt, Roberts and Sobel are tested for detection of the obstacles like trees, stones, etc., aiming to select which algorithm fits best for obstacles detection inside of the agricultural fields.

## **1.2 Contents of Sections**

The next Chapter gives an overview of vision system important in the agricultural robots. Chapter 3 explains the edge detection algorithms such as: Sobel, Canny, Prewitt and Roberts, which have been compared for the best performance in detecting agricultural obstacles in satellite pictures. Chapter 4 explains the properties of image samples and the application of algorithms by using Matlab. The results of the algorithms have been compared in the same chapter. Chapter 5 is reserved to the overall conclusion of the study.

## **Chapter 2**

### **AGRICULTURE ROBOT AND IMAGE PROCESSING**

Robots have been designed to perform their specific programmed tasks repetitively. Robots are created as an Artificial Intelligence and in order to move in real environment they need to have a visual system. For this purpose, scientist have introduced the Image processing techniques for robots to perform image analysis and to extract the important information for further information processing to perform their assigned tasks. The Edge and color detection techniques are used to strengthen vision system of robots so they can interact with environment and to deal with the objects effectively.

The vision system makes robots to be more efficient and effective and more in different fields such as in medical robots are helpful in surgery and other robots like mobile robots are useful in working in an open environment, such as military robots, space robots and agriculture robots. These robots are built with GPS and video camera systems in their body, which are directly linked to the central system with an access to satellite images.

In the field of robotics different image processing methods are used as per requirements, few need only edge detection so they locate the black and white color point in image to recognize the shape, and other robots need all colors information with edge detection and location, so they can move and be able to act upon given

instructions, for instance agriculture robot developed to recognize the bad grass and weeds in the fields, and to kill insects with a dose of fertilizer and to perform required remedies. The future version of this robot will be able to pull weeds from the roots completely as humans do without using herbicides as shown in Figure 1. These mobile robots perform their task by using image processing to detect grass and weed comparing the current image with the old image. If they recognize bad grass and weeds they will pull it out. They need satellite image for the movement inside the field with several sensors, such as spectral light curtains, imaging systems, laser distance sensors, and 3-D time-of-flight cameras to measure the spectral plant parameters and morphological in rows of plants such as maize [5].



Figure 1: Agriculture Robot while recognizing the bad grass and weeds in the fields

Another type of the agriculture robot is called “Harvey”, It is developed for the purpose of harvesting the agriculture. Harvey designed to move the shrubs and trees planted in containers around the nurseries. Application of these robots reduces the production costs and repetitive tasks.



This kind of mobile robot requires navigation system for locomotion, for this function they use satellite image to detect the obstacles. Once the edge is detected and the obstacle is recognized, they save the location and the size of the obstacles. There are different techniques used for these purpose such as Canny, Prewitt, Roberts and Sobel. In this study we are going to compare techniques that conclude which one is better and effective in agriculture fields.

## **Chapter 3**

### **EDGE DETECTION**

Edge detection has vital role in detecting shapes in an image, Goal of the edge detection is to detect the edges of the regions with a color sharp color difference in a picture. Edge detection provides flexibility and especially it is useful to recognize properties of the location with high accuracy, distinctively in aerial photos taken from satellites. Without edge detection, it is troublesome to differentiate the shapes on an image. With the assistance of edge detection, we are able to analyze the image, and we can represent the lines or circles. With the detected edges, it becomes straightforward to differentiate and perceive the elements of the image. Accuracy of identification depends on the effectiveness of the algorithms to the specific cases. Many researches afforded to develop new techniques of edge detection to seek out best results on totally different applications.

The motivation behind this research aim to detection the edges is to develop higher understanding and batter illustration of the elements of an image through digital processing. Humans have ability to discriminate the elements by lines or circles.

### 3.2 Robert edge detection Technique

The Roberts Cross operator performs a straightforward, quick to work out, and 2-D abstraction gradient mensuration on a picture. Within the output every constituent values represent the calculable magnitude of the abstraction gradient of the input image at that time. The operator consists of a combine of 2×2 convolution kernels. One kernel is solely revolved by 90°

$$G_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \text{ and, } G_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \quad (3.1)$$

These kernels area is unit designed to response to edges running at 45° to the constituent grid; one kernel is for 2 perpendicular orientations. To form separate measurements of the gradient part in every orientation (call these  $G_x$  and  $G_y$ ), kernels is applied severally to the input image. These will mix along to search out absolutely the magnitude and orientation of the gradient at every purpose. The gradient magnitude is provided by:

$$|G| = \sqrt{G_x^2} + \sqrt{G_y^2} \quad . \quad (3.2)$$

Although generally, Associate in Nursing approximate magnitude is computed using:

$$|G| = |G_x| + |G_y| \quad , \quad (3.3)$$

it is a lot of quicker to figure the angles of orientation increase the abstraction gradient (relative to the component grid orientation) that is:

$$\theta = \arctan (G_y/G_x) - 3\pi / 4 \quad (3.4)$$

### 3.3 Sobel Edge Detection Techniques

Standard Sobel announced operators for a  $3 \times 3$  neighborhood defines that every simple central gradient estimate is the resultant of a handful of orthogonal vectors [6]. These orthogonal vectors are basically the directional derivative estimate multiplied by a unit vector specifying the derivative's direction. These easy gradients add estimates quantity to a resultant of approx. 8 directional derivative vectors. Therefore, for a degree on mathematician grid and its eight neighbors having density values are shown as:

$$G = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}, \quad (3.5)$$

In directional by-product calculable vector, G was outlined as density distinction / distance to neighbor. This vector make up my mind because the direction of G, which can incline by the unit vector to the approximate neighbor. Purpose to be noted that the neighbors cluster into antipodal pairs: (a,i),(b,h),(c,g),(f,d). This gradient estimated is:

$$G = \frac{(c-g)}{R} \cdot \frac{[1,1]}{R} + \frac{(a-i)}{R} + (b-h) \cdot [0,1] + (f-d) \cdot [1,0], \quad (3.6)$$

where  $R = \sqrt{2}$  this vector is obtained a

$$G = [(c-g-a+i)/2 + f-d, (c-g+a-i)/2 + b-h] \quad (3.7)$$

Here, this vector is increased by a pair of attributable to commutation the divide by a pair of. The resultant formula is given as follow:

$$G = [(c - g - a + i) + 2.(f - d), (c - g + a - i) + 2.(b - h)]. \quad (3.8)$$

Weight functions for x and y parts were obtained by mistreatment the higher than vectors are mention as follows matrix:

$$x = \begin{bmatrix} 1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & 1 \end{bmatrix} \quad (3.9)$$

Now, we have a tendency to justify the dimension of the matrices, extended by exploitation [6]. The definition of the gradient will be used for 5×5 neighborhood [7]. During this case, twelve directional gradients should be determined rather than four gradients. Following matrix has 5×5 neighborhood:

$$G = \begin{bmatrix} a & b & c & d & e \\ f & g & h & i & j \\ k & l & m & n & o \\ p & q & r & s & t \\ u & v & x & y & z \end{bmatrix} \quad (3.10)$$

The resultant vector  $G'$  (like as the determination of Sobel 3×3 method) for 5×5 is given as follows:

$$G' = [20(n - l) + 10(i - r - g + t + o - k) + 5(e - v - a + z) + 4(d - w - b + y) + 8(j - p - f + u), 20(h - s) + 10(i - r + g - t) + 5 \cdot (e - v + a - z) + 4(j - p + f - u) + 8(d - w + b - y)] \quad (3.11)$$

The horizontal and vertical masks are obtained by using the coefficients in this equation.

$$M1 = \begin{bmatrix} -5 & -4 & 0 & 4 & 5 \\ -8 & -10 & 0 & 10 & 8 \\ -10 & -20 & 0 & 20 & 10 \\ -8 & -10 & 0 & 10 & 8 \\ -5 & -4 & 0 & 4 & 5 \end{bmatrix} \quad M2 = \begin{bmatrix} 5 & 8 & 10 & 8 & 5 \\ 4 & 10 & 20 & 10 & 4 \\ 0 & 0 & 0 & 0 & 0 \\ -4 & -10 & -20 & -10 & -4 \\ -5 & -8 & -10 & -8 & -5 \end{bmatrix} \quad (3.12)$$

These masks are used by the edge detection function in this section.

### 3.4 Prewitt Edge Detection Techniques

The operator consists of a combination of 3×3 convolution kernels as shown in the following matrix. One kernel is just the opposite turned by 90°.

$$Gx = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, Gy = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad (3.13)$$

These vertically and horizontally relative to the constituent grid, one kernel is for every of the 2 perpendicular orientations. In order to provide distinct measurements of the gradient part in every orientation (call these Gx and Gy), kernels is applied one by one to the input image. These are combined along to figure out the absolute the magnitude and orientation of the gradient for every purpose [8]. The gradient magnitude is given by:

$$|G| = \sqrt{Gx^2 + Gy^2} \quad (3.14)$$

Typically, AN approximate magnitude is computed by using:

$$|G| = |Gx| + |Gy| \quad (3.15)$$

It is work out as apace and therefore the angle of orientation of the sting (relative to the constituent grid) will increase the spatial gradient that is:

$$\theta = \arctan(Gy/Gx) \quad (3.16)$$

### 3.5 Canny Edge Detection Techniques

The clever edge detection formula is additionally referred to as the best edge detector.

Canny's formula is extremely useful in improving the performance of edge detectors [9].

To improve current strategies of edge detection, the list of criteria is mentioned below:

1. The primary and most evident expectation is the lowest error rate. The edges that occur in pictures must not be incomprehensible and it mustn't answer non-edges..
2. The second criterion is that the edge points should be localized. In different words, the gap between the edge pixels as found by the detector and also by the actual edge ought to be reduced.
3. A third criterion is one response to one edge. It has been enforced as a result of the primary 2 was not substantial enough to fully eliminate the likelihood of multiple responses to a feature.

Based on higher than criteria, the smart edge detector firstly smoothen the image by eliminating the noise and then finds the image gradient to spotlight regions with high spatial derivatives. The algorithmic program then tracks on that regions and suppresses any component that is not at most the utmost {the most} (non- maximum suppression). The gradient array is currently any small by physical phenomenon. Physical phenomenon is beneficial to trace on the remaining pixels that haven't been hid. Physical phenomenon typically uses 2 thresholds and if the primary threshold has low magnitude, it's set to zero. If the magnitude is higher than the high threshold, it then this creates a feature and if the magnitude lies between the 2 thresholds, then it sets to zero unless it gets a path from this component to a gradient that is higher than  $T_2$ .

**Step 1:** In order to implement the clever edge detector formula, number of of steps should be performed. First of all the first image would be filtered out before attempting to find and sight of any edges. The mathematician filter will reckon employing an easy mask, it will be used completely within the clever formula. Presently the appropriate mask would be calculated and then mathematician smoothing is performed by victimization commonplace convolution ways. A convolution mask is typically smaller than the particular image. As an output, the mask is glide over the image, and manipulates a sq. of pixels at a time. Because the larger dimension of the mathematician mask, it lowers the detector's compassion to noise. The localization error within the known edges additionally is enlarged to a small degree because the mathematician dimension is enlarged.

**Step 2:** After smoothing and eliminating the noise of the image, future step is to search out the sting strength by taking the gradient of the image. A 2-D spatial gradient measure would be performed by the Sobel on the image. After that, the calculable absolute gradient scale at every purpose is found. The Sobel operator [9] uses a combine of 3x3 convolution masks, in one estimates the gradient within the x-direction (columns), and, also, the alternative estimates the gradient within the y-direction (rows). As shown below:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (3.17)$$

The magnitude or edge strength of the gradient is then measured by the exploitation the subsequent formula:

$$|G| = |G_x| + |G_y| \quad (3.18)$$



**Step 3:** The direction of the sting is measured by the victimization gradient within the x and y directions, then a slip are going to be generated once total X would be up to zero. Thus, within the code there ought to be a restriction set once ever this happens and when the gradient within the x direction is up to zero. The sting direction ought to be up to ninety degree or zero degree depending upon what the worth of the gradient within the y-direction is up to. If GY is up to worth of zero, the sting direction than will be up to zero degrees, whereas the sting direction are going to be up to ninety degree. To search out the sting direction required following formula is states below:

$$\text{Theta} = \tan^{-1} (\text{Gy} / \text{Gx}) \quad (3.19)$$

**Step 4:** Once the sting direction is founded, the final step is to relate the sting direction to a direction that may be derived in a picture. Therefore the pixels of a 5x5 image are aligned as follows:

```

X X X X X
X X X X X
X X a X X
X X X X X
X X X X X

```

It will be determined at component "a", there are a unit solely four potential directions that describes the encircling pixels - zero degrees (in the horizontal direction), forty five degrees (along the positive diagonal), ninety degrees(in the vertical direction), or one hundred thirty five degrees (along the negative diagonal). Hence, the sting orientation currently should be resolved into one among these four directions counting on that direction is nearest to (e.g. if the orientation angle is found to be three degrees, build it zero degrees). It divides semi-circle into five regions it is shown in Figure 2

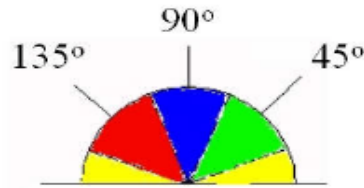


Figure 2: Semi-circle into 5 regions

Therefore, the sting direction falling at intervals shown as the yellow vary (0 to twenty two.5 & 157.5 to one hundred eighty degrees) is ready to zero degrees. The sting direction falling within the inexperienced vary (22.5 to 67.5 degrees) is ready to forty five degrees. the sting direction falling within the blue vary (67.5 to 112.5 degrees) is ready to ninety degrees. And lastly, the sting direction falling at intervals shown as the red vary (112.5 to 157.5 degrees) is ready to one hundred thirty five degrees.

**Step 5:** Once the sting directions square measure is determined, non-maximum suppression is applied. Non-maximum suppression is employed to trace on the sting within the edge direction and suppress any component worth (sets it up to 0) that does not contemplate as a grip. It provides a skinny line in output image.

**Step 6:** Finally, hysteresis [10] is used as a means of eliminating the streaking. Streaking is the breaking up of an edge contour caused by the operator output fluctuating above and below the threshold. If a single threshold, T1 is applied to an image, and an edge has an average strength equal to T1, then due to the noise, there will be instances where the edge dips will be below the threshold. Equally it will also extend above the threshold making an edge look like a dashed line. To avoid this, hysteresis uses 2 thresholds, a high and a low thresholds. Any pixel in the image that has a value greater than T1 is presumed to be an edge pixel, and is marked as immediately. Then, any pixels that are connected to this edge pixel and that have a value that is greater than T2 are also selected as edge pixels. If you think of

following an edge, you need a gradient of T2 to start but you do not stop till you hit a gradient below T1.

## Chapter 4

### EDGE DETECTION TECHNIQUES ON THE FIELD IMAGES

This chapter compares different edge detection techniques (Canny, Prewitt, Roberts and Sobel) in order to detect an obstacle in the agricultural field images. An obstacle refers to the image of a tree in the field. The Matlab Image Processing application is employed to apply the edge detection techniques on the field images.

#### 4.1 Obtaining satellite image and edge detection

The initial stage of the system is to import satellite image. Our system benefits from Google Maps image API V2 [11] to locate and import satellite images using an API Key similar to Figure. 3.

```
http://maps.googleapis.com/maps/api/staticmap?center=New+York,NY&
zoom=13&size=600x300&key=API_console_key
```

Figure 3: Sample API key is used to obtain Satellite images from Google maps.

While determining the boundaries of the desired field, the highest possible zoom level ( $\Delta_z$ ) is selected to fit the whole field in the image frame. Google provides different range of zoom levels that is usually range from zero to twenty-two, zero being the minimum, which fits the whole earth in the picture, and the highest zoom levels providing more details. For each field, the highest possible zoom level, which could fit the field with maximum details, is determined manually to initialize automatic image processing in the image processing part.

## 4.2 Image processes on the field image Extraction

Median filtering is a nonlinear operation that is used to eliminate sharp singular marks, which are out of interest due to their small size. Median filtering is preferred to other filters due to its potential to preserve edges while suppressing noise. Conserving edges are necessary to extract the field and obstacles successfully. Some segmentation techniques [12] along with “regioning” are used to extract desired agricultural field from the imported image. After converting imported image to gray-scale, a binary form of the image is constructed that replaces all values in the image matrix by zero or one indicating a black or white pixel respectively [13] Figure 4 b. Demonstrates the image in binary form gives us the opportunity to find regions in addition to edges. A region is determined by the connected pixels of the same value.

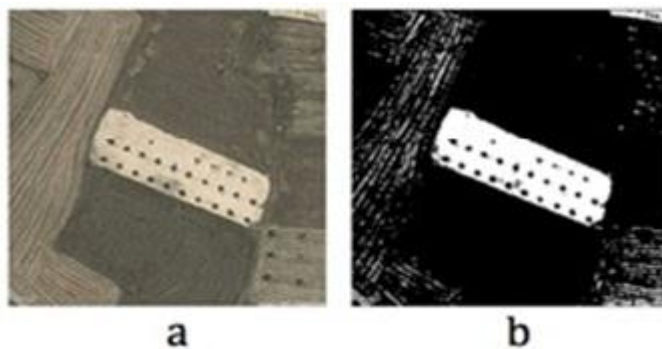


Figure 4: a) Original image b) Binary image,

#### 4.4 Image processing to detect obstacle(s)

The binary (gray-scale) image obtained is from the previous section that is used to detect objects. Although this image includes desired field, there might be some spurious noises inside the image as shown with arrows in Figure 5. This kind of noise adds complication to detection of obstacles. To overcome this problem, we reconstructed this image from the largest stored region. This eliminates the objects outside the field and makes the image to contain only the desired field as it seen in Figure 5b.

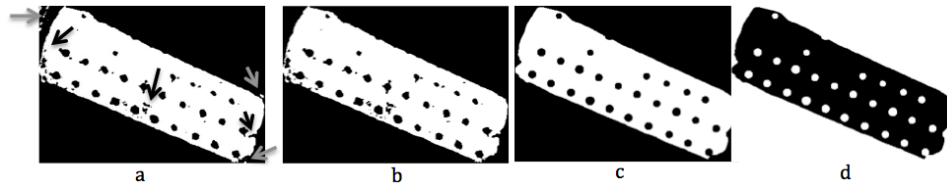


Figure 5: a) Binary image with undesired segments, b) undesired segment are removed from outside, c) desired field free off undesired objects, d) inverted image

The undesired regions, which were inside of the field however too small to be an obstacle, it was removed by using the algorithm based on morphological image reconstruction by filling them [14]. This process resulted in a clearer, smoother and more accurate image as it is shown in Figure 5c. The obstacle detection algorithm inverts the resulted binary image. The outcome is a picture with white regions on black field [15], as shown in Figure 5d. This inversion prepares the image for the segmentation technique, this time for determining the obstacles [16].

Detected regions are trees and their attributes are to be obtained by using the Circular Hough Transform algorithm, which detects all of the circular shaped objects inside the field [17]. Typically, at zoom level eighteen, trees have radius range from four to eighteen pixels. A circular object must have a suitable size for a tree, and must be in

the region of the field. Once the circular shape passes the condition, its center point and radius in pixels are saved to the list of objects and attributes.

Once the circular shape passes the condition, its center point and radius in pixels is saved to the list of objects and attributes. The algorithms was applied in Matlab program to be tested, as it is shown in Figure 6 for Block diagram show the Matlab steps processing for edge detection techniques on 42 different fields with various shapes, sizes and number of obstacles to check the system.

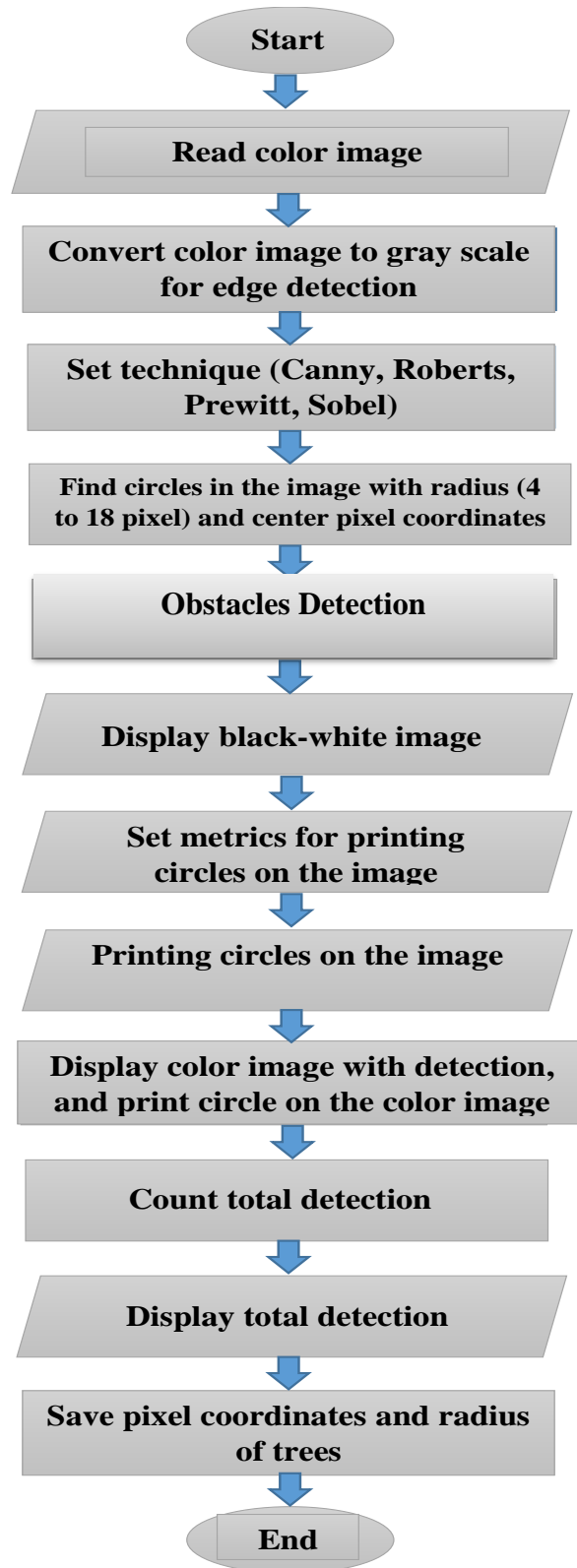


Figure 6: Block diagram showing the Matlab steps processing for edge detection techniques



## 4.5 Error evaluation

In the experience process, the evaluation of each technique (Canny, Prewitt, Roberts and Sobel) has two main types of errors, first one called the false positive errors which represents miss obstacles detection errors inside the fields, which mean that there is a tree but the technique cannot detect it. The second is the false negative error which represents wrong detection on empty space inside the field, these techniques locate tree on an empty space while in fact there are no trees in that location.

## 4.6 Results and Discussion

In order to demonstrate, which technique (Canny, Robert, Prewitt and Sobel) is the most efficient one a number of examples are available. The algorithms was tested in Matlab program (as shown previously in Figure 6 for each techniques on 42 different fields with various shapes, sizes and number of obstacles to check the system. Some of the fields are presented in Figure 7 which has different complexity.

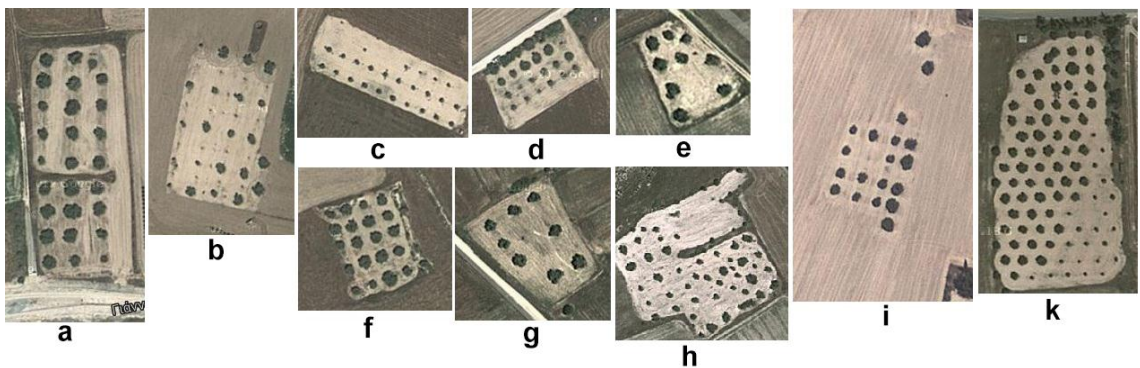


Figure 7 : Ten Test Fields with Different Size, Shape and Complexity

The zoom level in all tests is set to 18 for comparison. Also, the radius range for detection of trees is set to  $mi=4$  and  $max=18$  in pixels. In addition, tree detection

outputs of these fields were graphically presented, which each red circle on the images denotes a recognized tree:

1-Canny technique: The results of 1282 out of 1602 (80%) are detected obstacles correctly. It has false positive error that represents 323 (20%) miss obstacles detection errors inside the fields b2, c2, d2, f2, e2, h2, i2 and false negative error represents 283 (20%) wrong detection on empty space inside the field a2, b2, c2, e2, f2, h2, i2 total errors for this technique is 40% as it is shown in Figure 8. Ten samples of the fields the binary image gray-scale in the top show the edge detection of fields and down the color image for the fields. For all of the samples tested in canny technique for false positive errors (F.P.E) is shown in Figure 9 and Figure 10 for false negative error (F.N.E) for all the samples.

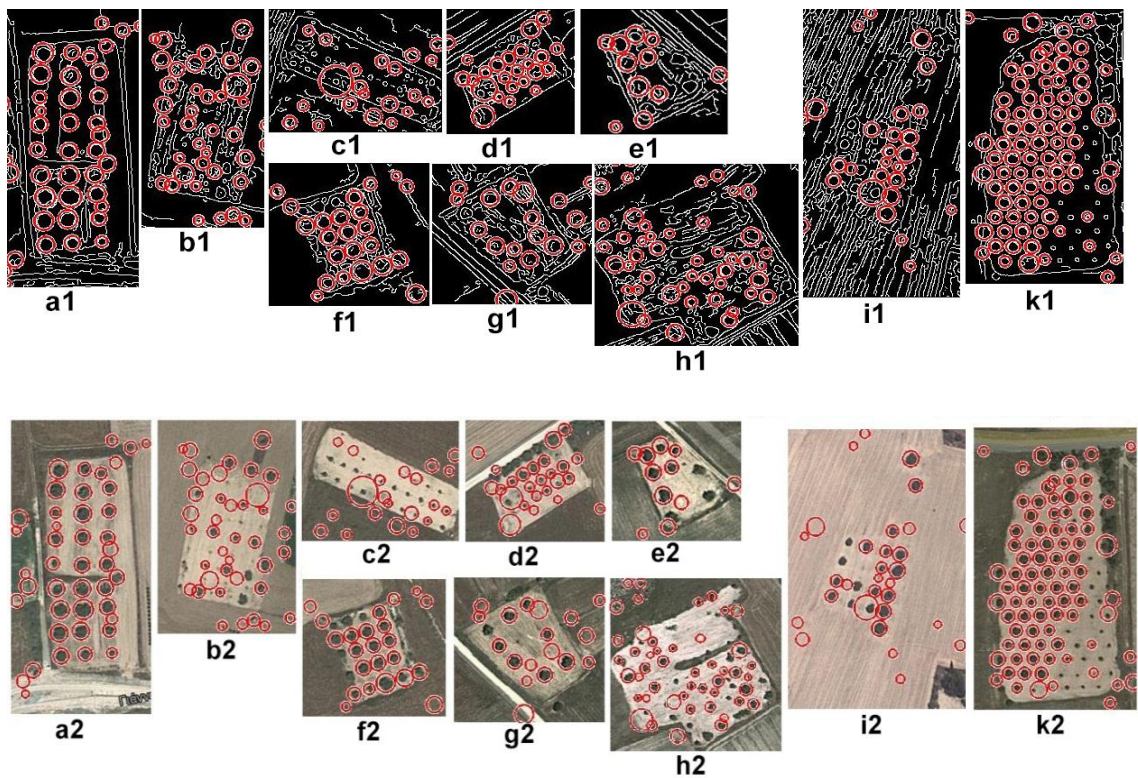


Figure 8: Canny edge detection technique

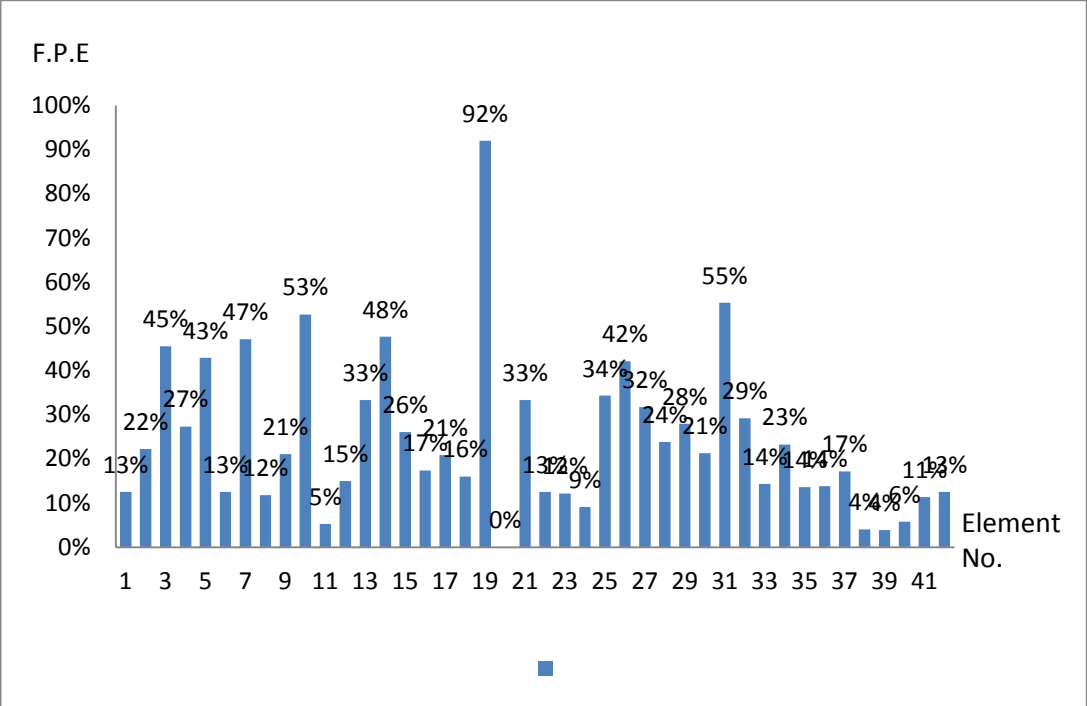


Figure 9: Canny edge detection technique for all samples false positive errors 20%

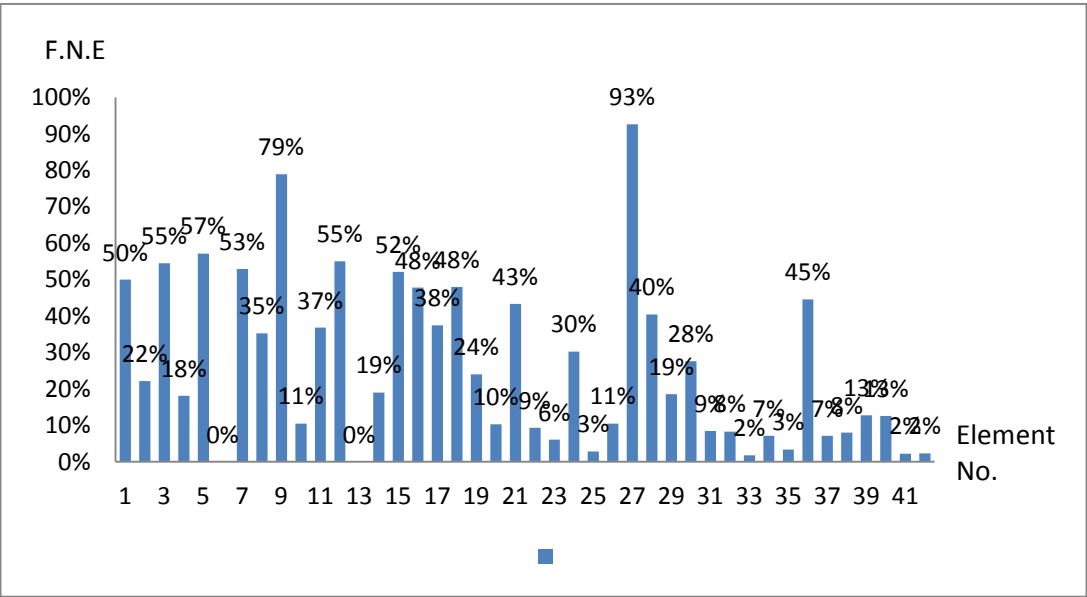


Figure10: Canny edge detection technique for all samples false negative errors 20%

2-Prewitt edge detection technique: the results of 1333 out of 1602 (83%) were correctly detected. It has false positive error that represents 283 (17%) miss obstacles detection inside the fields a2, c2, d2, f2 and false negative error represents 155(9%)wrong detection inside the fields a2, g2, k2 the error , total errors for this technique is 26% as it displayed in the Figure 11. Ten samples of the fields are the binary image gray-scale in the top that shows the edge detection of the fields and down the color image for the fields. For all of the samples tested in Prewitt technique false positive errors (F.P.E) is shown in Figure 12 and Figure 13 for false negative error (F.N.E) for all samples, sample number 21 has out layer value of 30 obstacles and it found 71 wrong obstacles detection is found.

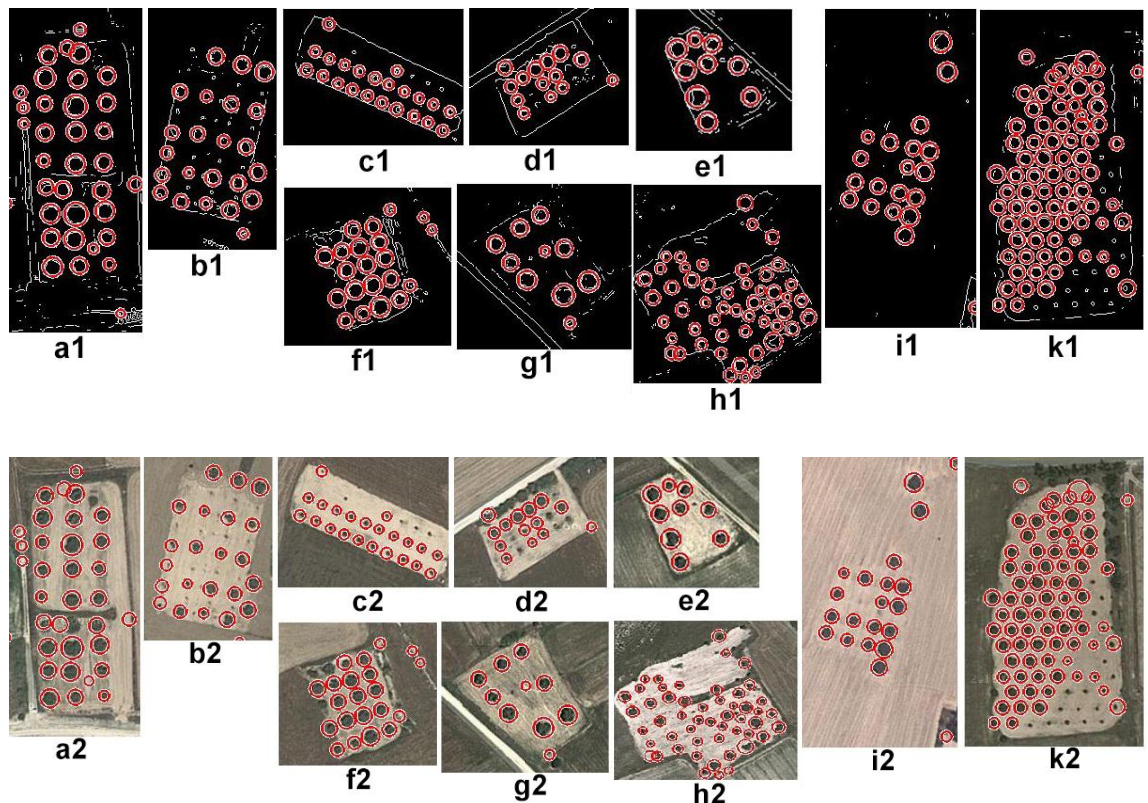


Figure 6: Prewitt edge detection technique

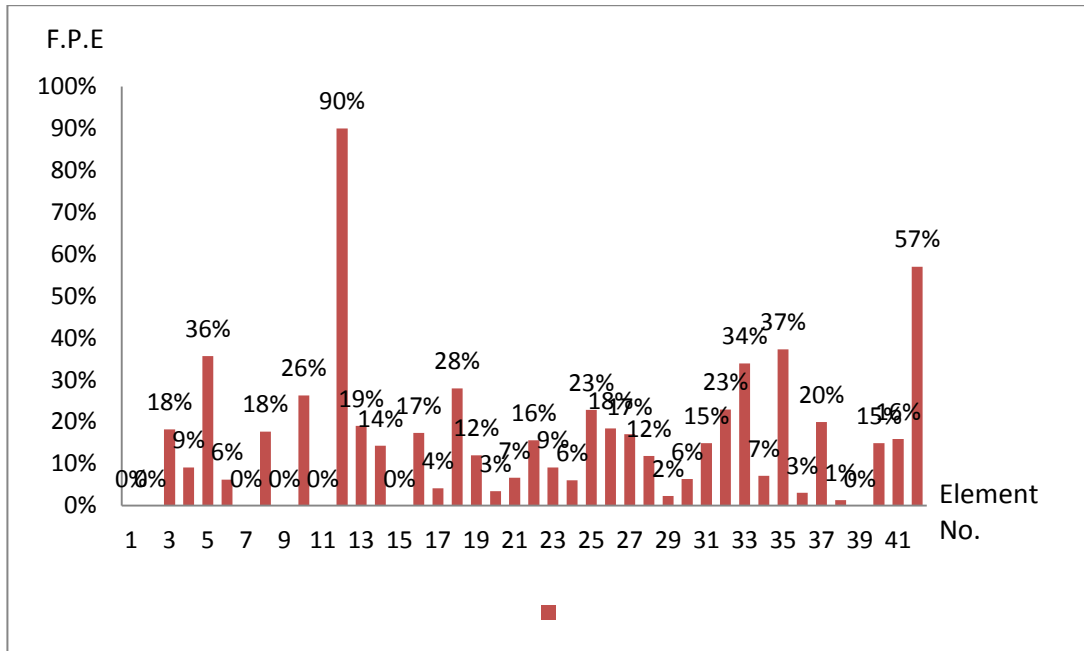


Figure 7: Prewitt edge detection technique for all samples false positive errors 17%

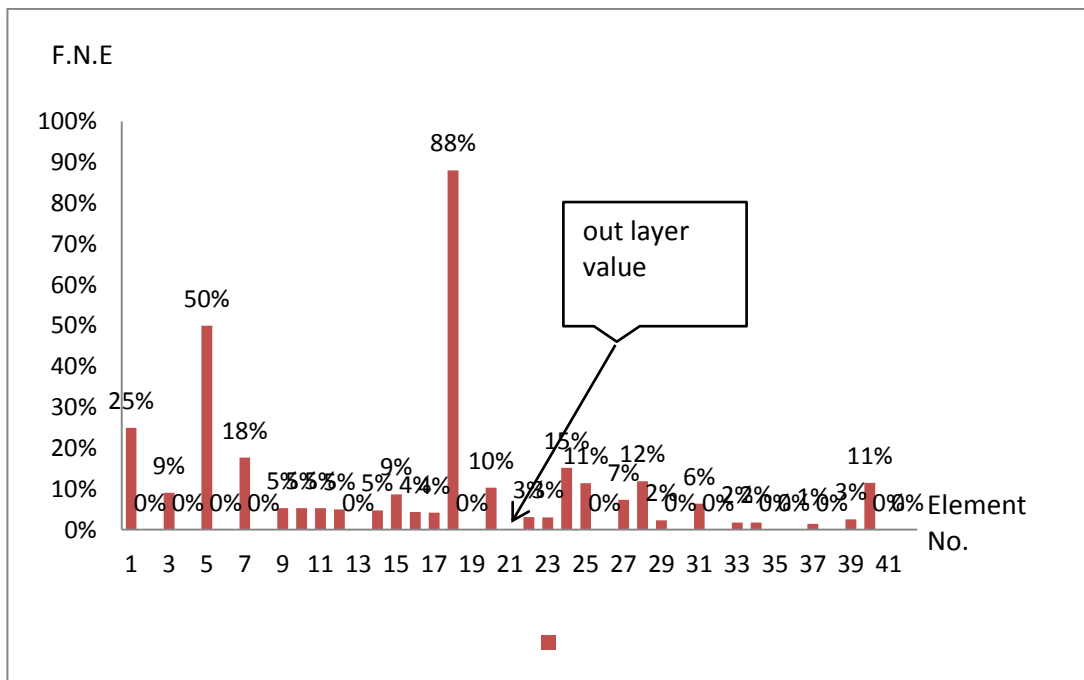


Figure 8: Prewitt edge detection technique for all samples false negative errors 9%

3- Roberts edge detection technique, the results of 1044 out of 1602 (65%) correctly Obstacles detection, it has the false positive error that represents 557 (35%) miss obstacles detection inside the fields of a2, c2, d2, e2, f2, h2,k2 and false negative error represents 105(6.5%) wrong detection inside the fields a2, b2, f2, g2, h2 , total errors for this technique is 41.5% as it is demonstrated in the Figure 14. Ten samples of the fields are the binary image gray-scale in the top show the edge detection of fields and down the color image for the fields. for all samples tested in Roberts technique false positive errors (F.P.E) shown in Figure 15 and Figure 16 for false negative error (F.N.E) for all samples, sample number 21 have out layer value it have 30 obstacles and it found 50 wrong obstacles detection.

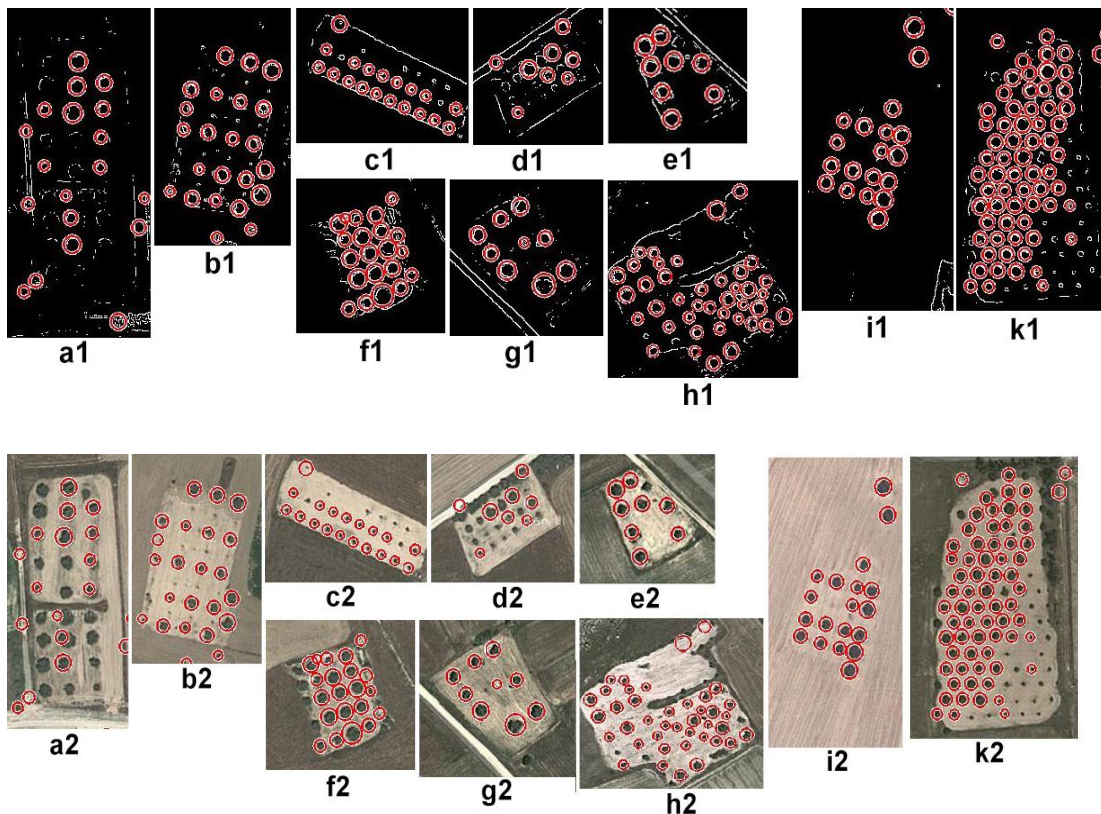


Figure 9: Robert edge detection technique

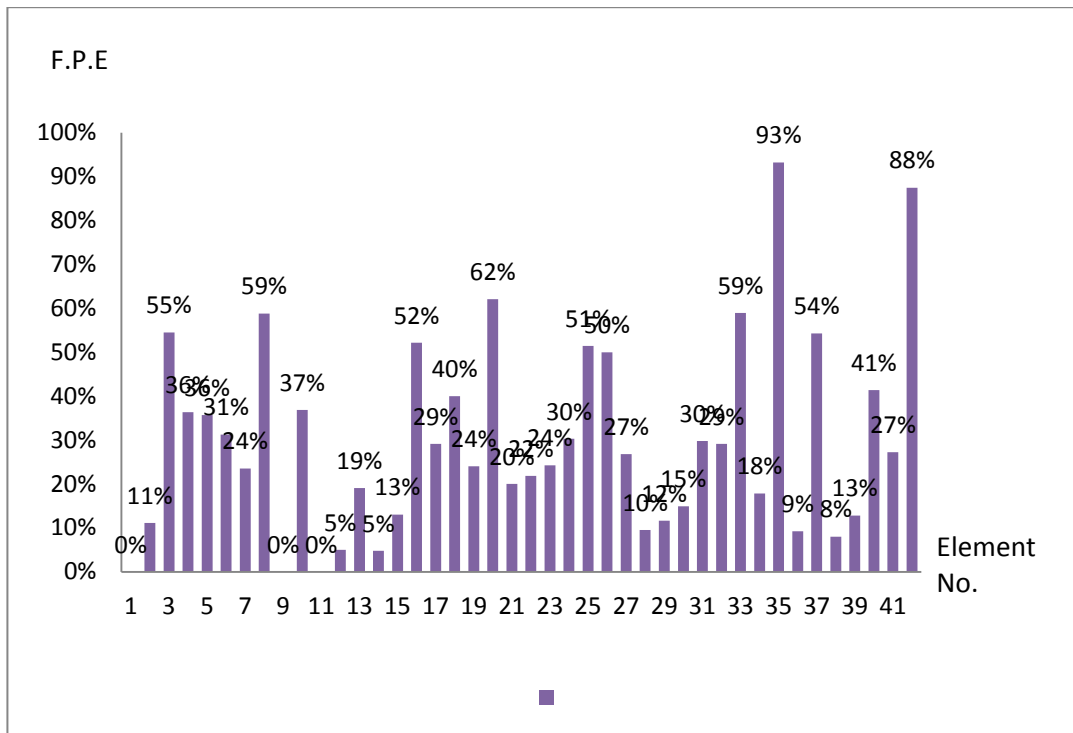


Figure 10: Robert edge detection technique for all samples false positive errors 35%

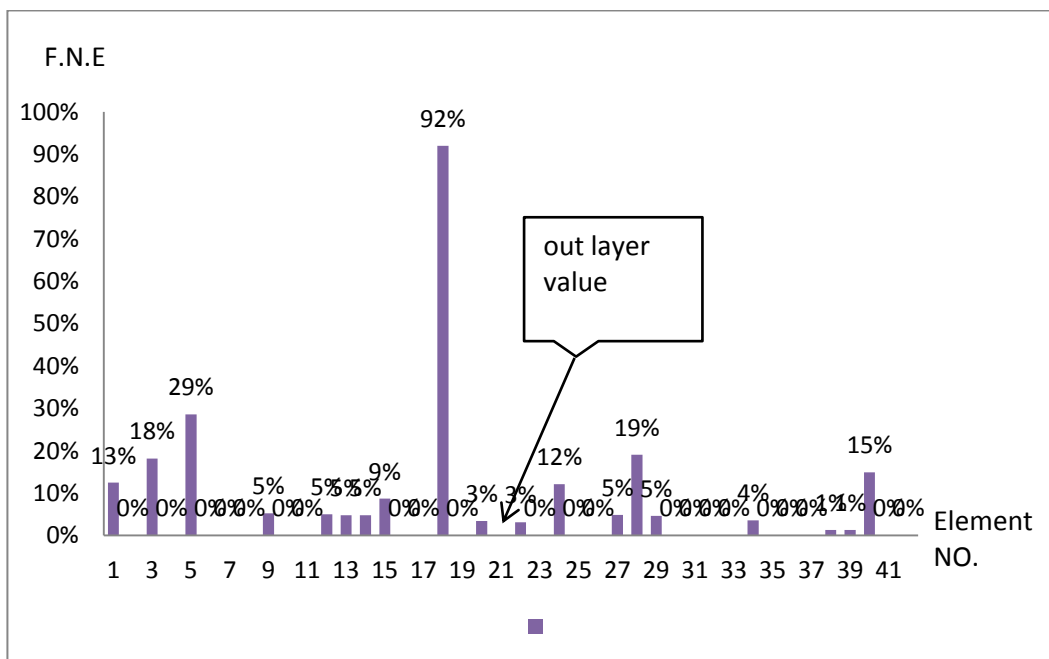


Figure 11: Robert edge detection technique for all samples false negative errors 6.5%

4- Sobel edge detection technique: the results of 1350 out of 1602 (84%) correct detection. The errors are 253 (16%) miss obstacles detection inside the field, as it is shown in the fields of a2, c2, d2 and 106(6.5%) wrong detection inside the field of g2 the error. Total errors for this technique are 22.5% shown in the Figure 17. Ten samples of the fields are the binary image gray-scale in the top that shows the edge detection of fields and down the color image for the fields. For all samples tested in Sobel technique false positive errors (F.P.E) is shown in Figure 18 and Figure 19 for false negative error (F.N.E) for all samples.

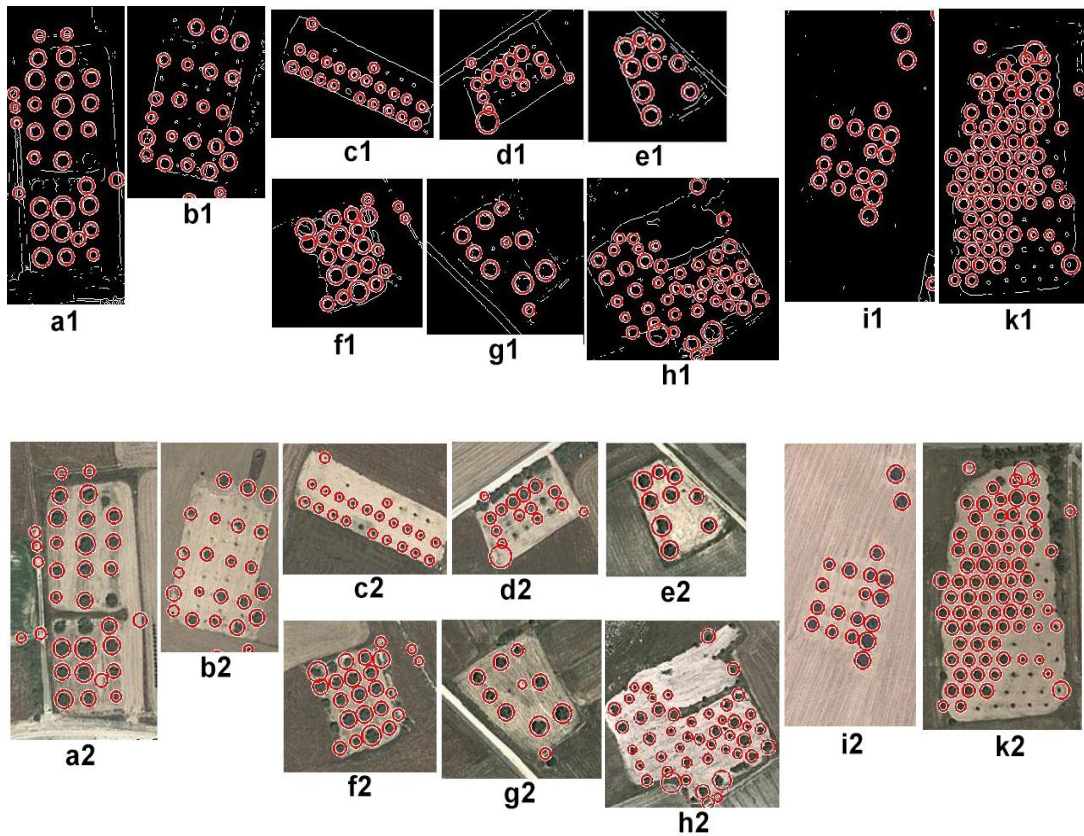


Figure 17: Sobel edge detection technique



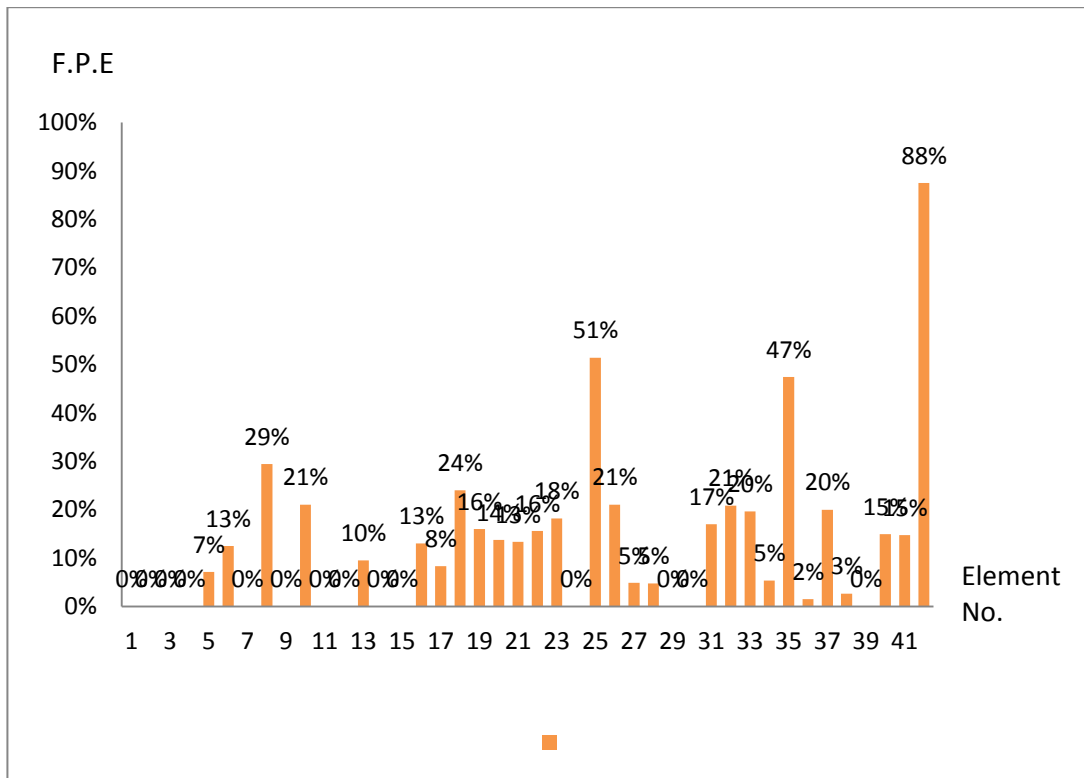


Figure 18: Sobel edge detection technique for all samples false positive errors 16%

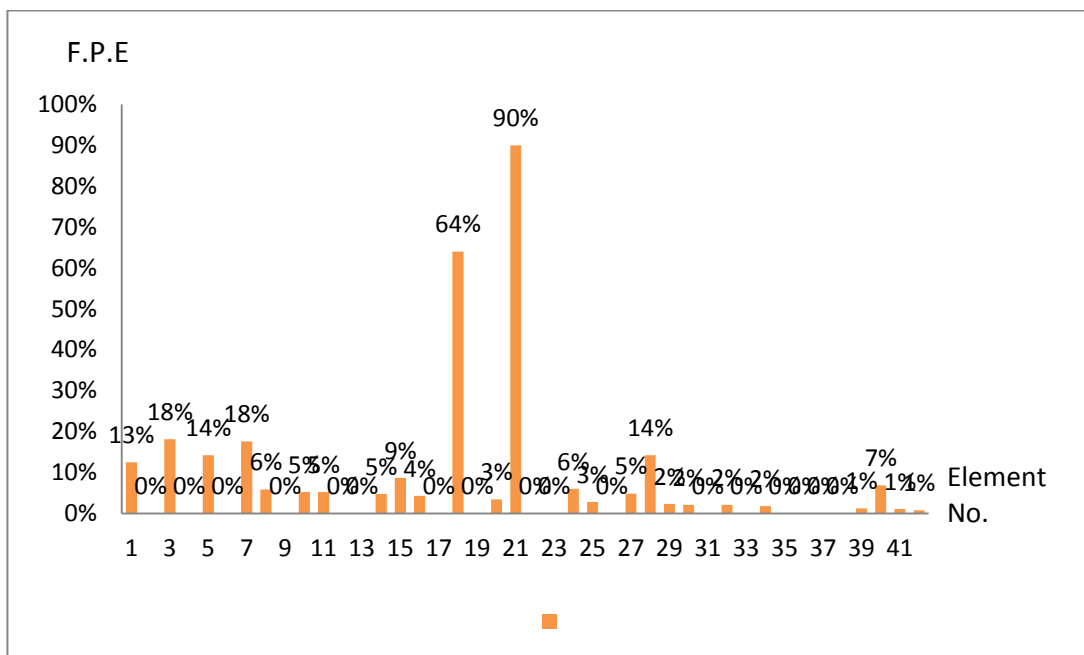


Figure 19: Sobel edge detection technique for all samples false negative errors 6.5%

Figure 20 shows all of the techniques for all the samples with false positive errors (F.P.E) and false negative errors (F.N.E), Canny technique in blue color, Prewitt technique in red color, Roberts technique in Purple color and Sobel technique in orange color.

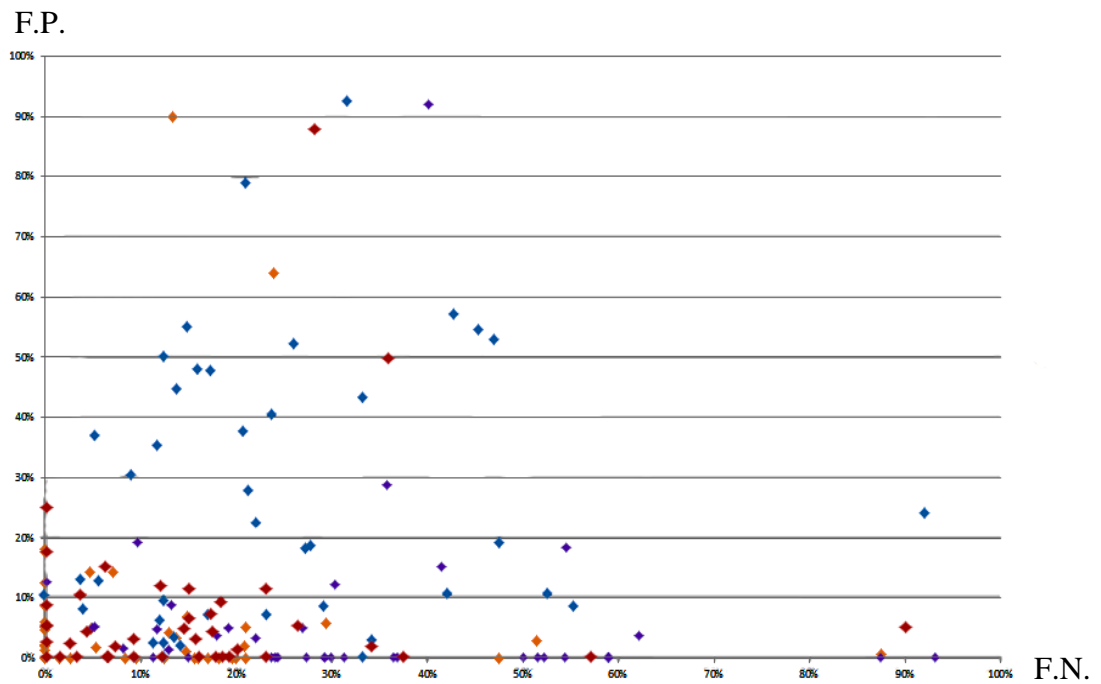


Figure 20: All techniques for all of the samples false negative errors and false positive errors

Details of the detection section for these 41 fields are shown in Table. 4.1. First column represents the total number of the samples (NO.) and the second column shows the total number of trees existing in the field (N.T), whereas third displays the correct detections by using the canny technique (C.C.H), the fourth column shows the correct detections by using the Prewitt technique (P.C.H), fifth column represents the correct detections by using the Robert’s technique (R.C.H), and the last column shows the correct detections by using Sobel technique (S.C.H)

Table 4.1: Data regarding correct obstacle detection section

NO.	N.T	C.C.H	P.C.H	R.C.H	S.C.H
1	8	7	8	8	8
2	9	7	9	8	9
3	11	6	9	4	11
4	11	8	10	7	11
5	14	8	9	9	13
6	16	14	15	11	14
7	17	8	17	13	17
8	17	15	14	7	12
9	19	9	14	12	15
10	19	15	19	19	19
11	19	18	19	19	19
12	20	17	20	19	20
13	21	16	17	17	19
14	21	11	18	20	21
15	23	17	23	20	23
16	23	19	19	11	20
17	24	19	24	17	23
18	25	6	22	19	21
19	25	21	18	15	19
20	29	29	28	11	25
21	30	20	28	24	26
22	32	28	27	25	28
23	33	30	31	23	33
24	33	29	30	25	27
25	35	23	17	17	17
26	38	24	31	19	30
27	41	28	34	30	38
28	42	32	37	38	40
29	43	31	42	38	43
30	47	21	40	33	39
31	47	37	45	40	47
32	48	34	37	34	38
33	56	48	37	23	45
34	56	43	52	46	53
35	59	51	37	4	31
36	65	56	63	59	64
37	70	58	56	32	56
38	75	72	74	69	73
39	78	75	78	68	78
40	87	82	74	51	74
41	88	78	74	64	75
42	128	112	57	16	56
Total	1602	1282	1333	1044	1350
percent correct		80%	83%	65%	84%

Details of the detection section for these 42 fields are shown in Table. 4.2. The first column represents the total number of samples (NO), the second column shows the number of obstacles (trees) in the fields (N.T), third column demonstrates the canny technique miss obstacles detection errors (C.E.M), fourth column indicates the canny technique wrong obstacles detection errors (C.E.W.), fifth column presents the Prewitt technique miss obstacles detection errors (P.E.M), sixth column shows the Prewitt technique wrong obstacles detection errors (P.E.W), seventh column is about the Roberts technique miss obstacles detection errors (R.E.M), eighth column shows the Roberts technique wrong obstacles detection errors (R.E.W.), ninth column demonstrates the Sobel technique miss obstacles detection errors (S.E.M), and the last column represents the Sobel technique wrong obstacles detection errors (S.E.W.M).

Table 4.2: Data regarding missed and wrong obstacle detection section

NO.	N.T.	C.E.M.	C.E.W.	P.E.M.	P.E.W.	R.E.M.	R.E.W.	S.E.M.	S.E.W.
1	8	1	4	0	2	0	1	0	1
2	9	2	2	0	0	1	0	0	0
3	11	5	6	2	1	6	2	0	2
4	11	3	2	1	0	4	0	0	0
5	14	6	8	5	7	5	4	1	2
6	16	2	0	1	0	5	0	2	0
7	17	8	9	0	3	4	0	0	3
8	17	2	6	3	0	10	0	5	1
9	19	10	2	5	1	7	0	4	1
10	19	4	15	0	1	0	1	0	0
11	19	1	7	0	1	0	0	0	1
12	20	3	11	20	1	1	1	0	0
13	21	7	0	4	0	4	1	2	0
14	21	10	4	3	1	1	1	0	1
15	23	6	12	0	2	3	2	0	2
16	23	4	11	4	1	12	0	3	1
17	24	5	9	1	1	7	0	2	0
18	25	19	6	3	0	6	0	4	0
19	25	4	12	7	21	10	7	6	16
20	29	0	3	1	3	18	1	4	1
21	30	10	13	2	71	6	50	4	50
22	32	4	3	5	1	7	1	5	0
23	33	3	10	2	5	10	4	0	2
24	33	4	2	3	1	8	0	6	0
25	35	12	1	8	4	18	0	18	1
26	38	16	4	7	0	19	0	8	0
27	41	13	45	7	3	11	2	2	2
28	42	10	17	5	5	4	8	2	6
29	43	12	8	1	1	5	2	0	1
30	47	26	4	7	3	14	0	8	0
31	47	10	13	3	0	7	0	0	1
32	48	14	4	11	0	14	0	10	1
33	56	8	1	19	1	33	0	11	0
34	56	13	4	4	1	10	2	3	1
35	59	8	2	22	0	55	0	28	0
36	65	9	29	2	0	6	0	1	0
37	70	12	5	14	1	38	0	14	0
38	75	3	6	1	0	6	1	2	0
39	78	3	10	0	2	10	1	0	1
40	87	5	11	13	10	36	13	13	6
41	88	10	2	14	0	24	0	13	1
42	128	16	3	73	0	112	0	72	1
Total	1602	323	326	283	155	557	105	253	106
Percent. Errors		20%	20%	17.5%	9%	35%	6.5%	16%	6.5%

## **4.7 Determining the Location (Coordinates) and Size of the Obstacles**

The proposed system mainly works on top-view satellite images, which is processed by using the image-processing techniques to extract the field from the image [18]. The obtained image techniques are processed to detect the boundaries of the field, and the obstacles [19]. Different kinds of obstacles can exist in an agricultural field, the image techniques coordinates and radius for each obstacle inside the field as it is shown in table 4.3. As an example; all detected obstacles together with their tagged attributes are presented in an exchangeable format for further usage by subsystems or ontologies. Extended Markup Language (XML) is used as a standard to provide exchangeable data due to its capability of documenting data and information of both machine and human readable format [20], as shown in Figure 25, Mapping of fields coordinates to the center pixel of one obstacle in original image.



Figure 12: Mapping of field coordinates to the center pixel of obstacles in original image

Table 4.3: Data regarding obstacle coordinates and radius

NO.	X	Y	radius
1	153.9462	242.3186	9.143661
2	148	197	7.319793
3	95.44587	202.4804	7.615865
4	199.4671	67.21295	8.854716
5	101.5307	184.0404	7.196644
6	138.5713	143.0789	7.838514
7	115.2148	136.4875	6.075157
8	125.7871	189.6464	7.125062
9	107.1128	159.149	6.998922
10	174.9075	173.8122	8.786895
11	165.049	201.5518	8.767315
12	170.4851	123.9597	7.639218
13	179.6231	152.5127	9.206625
14	192.3027	36.85882	9.608296
15	156.8545	168.9678	6.439984
16	162.5275	148.2902	7.246589
17	141.4017	216.0644	7.150764
18	159.0971	221.1185	9.70268
19	118.4045	210.7712	6.690352
20	269.1274	325.0544	6.211589
21	257.6139	327.59	6.610759
22	229.2459	318.4889	6.090813
23	238.9788	14.01992	9.726698



## Chapter 5

### CONCLUSION

The main focus of this study was to compare different methods for obstacle detection, positioning and semantic annotation for automating agricultural systems for the automatic detection of the agricultural obstacles like trees mostly. Edge detection is essential in agriculture system automations. The locations of the detected obstacles are determined sufficiently in precise and accurate manner for agricultural processes. Along with simplicity of automated obstacle detection, the accuracy in positioning the obstacles makes the developed method feasible for real-time utilization of a subsystem. That subsystem is a part of an automated agricultural plantation system.

The edge detection techniques Canny, Prewitt, Roberts and Sobel are compared to detect the obstacles with the precision of free Google Maps API, which provides maximum 640 by 640 pixels image. The zoom level of free API is set to  $z=18$  for the satellite image to cover the complete agricultural field. The success rates of the detected trees by each method are compared to determine the performance of each edge detection method. Canny edge detection technique got 80% correct obstacle detection, making %20 false positive errors miss obstacles inside the fields. Furthermore, the Canny also has the false negative errors of 20% error of wrong obstacle detection inside the fields. Total error for this technique is 40%. Prewitt edge detection technique found 83% correct obstacle detection, and 17% false positive

errors miss obstacle detection error and it has 9% of false negative errors of wrong obstacle detection inside the fields. Total error for this technique is 23%. Roberts edge detection technique got 65% correct obstacle detection and 35% false positive errors miss obstacle detection error and it has 6.5% of false negative errors error of wrong obstacle detection inside the fields. Total error for this technique is 41.5%. Moving onto Sobel edge detection technique has found 84% correct obstacle detection and 16% of false positive errors miss obstacle detection error and it has 6.5% of false negative error of wrong obstacle detection inside the fields. Total error for this technique is 22.5%

Hence, according to our analyze we can state that the best technique is Sobel edge detection, it has higher average accuracy in detecting trees around 84% and lower percent misdetection false positive error is 16% also lower false negative error wrong obstacle detection inside the fields is 6.5% total errors for this technique is 22.5% from other techniques which represents a significant result, which is precise enough for any agricultural purpose.

According to future aspect, Sobel techniques will be more accurate to detect the obstacle and more flexible to work with different agricultural fields. They can even develop it to detect and found the desired agricultural field between neighboring fields and to differentiate obstacles such as stones.

## REFERENCES

- [1] Wikipedia, “Robot” [online]. Available: <http://en.wikipedia.org/wiki/Robot> [Accessed December 2013].
- [2] Modern farmer, “5 Coolest Farm Robots” [online]. Available <http://modernfarmer.com/2013/08/5-robots-on-the-farm/> [Accessed December 2013].
- [3] Mehmet Bodur, Ehsan Kiani, & Hasan Hacisevki, (2012). Double look-ahead reference point control for autonomous agricultural vehicles, biosystems engineering 113 173-186
- [4] Thomas Deselaers, Daniel Keysers, & Hermann Ney, (2005). Discriminative Training for Object Recognition Using Image Patches, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02 .
- [5] Vision Systems, “Robot plants itself in phenotyping task” [Online]. Available: <http://www.vision-systems.com/articles/slideshow/robots-join-forces-with-vision-systems/pg004.html> [Accessed December 2013].
- [6] Sobel, I., (1990). *An Isotropic 3×3 Gradient Operator*, Machine Vision for Three – Dimensional Scenes, Freeman, H., Academic Pres, NY, 376-379.

- [7] Aybar, E., (2003). “Topolojik Kenar \_slecleri”, Anadolu Üniversitesi, Fen Bilimleri Enstitüsü, Ph.D. thesis.
- [8] J. Matthews. (2002). “An introduction to edge detection: The sobel edge detector,” Available at <http://www.generation5.org/content/2002/im01.asp>.
- [9] J. F. Canny. (1986). “A computational approach to edge detection”. IEEE Trans. Pattern Anal. Machine Intell., vol. PAMI-8, no. 6, pp. 679-697.
- [10] J. Canny. (1983). “Finding edges and lines in image”. Master’s thesis, MIT.
- [11] Static Maps API V2 Developer Guide. Google Inc. Web. 11 July. 2013. <https://developers.google.com/maps/documentation/staticmaps/>
- [12] Muhammad Rizwan Khokher, Abdul Ghafoor, & Adil Masood Siddiqui, (2013). Image segmentation using multilevel graph cuts and graph development using fuzzy rule-based system, IET Image Process., Vol. 7, Iss. 3, pp. 201–211 201
- [13] Anuj Mohan, Constantine Papageorgiou, & Tomaso Poggio, (APRIL 2001). Example-Based Object Detection in Images by Components, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 23, NO. 4, Pages 349—361.
- [14] Luc Vincent, (April 1993). Morphological Grayscale Reconstruction in Image Analysis: Applications and Efficient Algorithms. IEEE Transactions on Image Processing, Vol. 2, No, 2, pp.176-201.

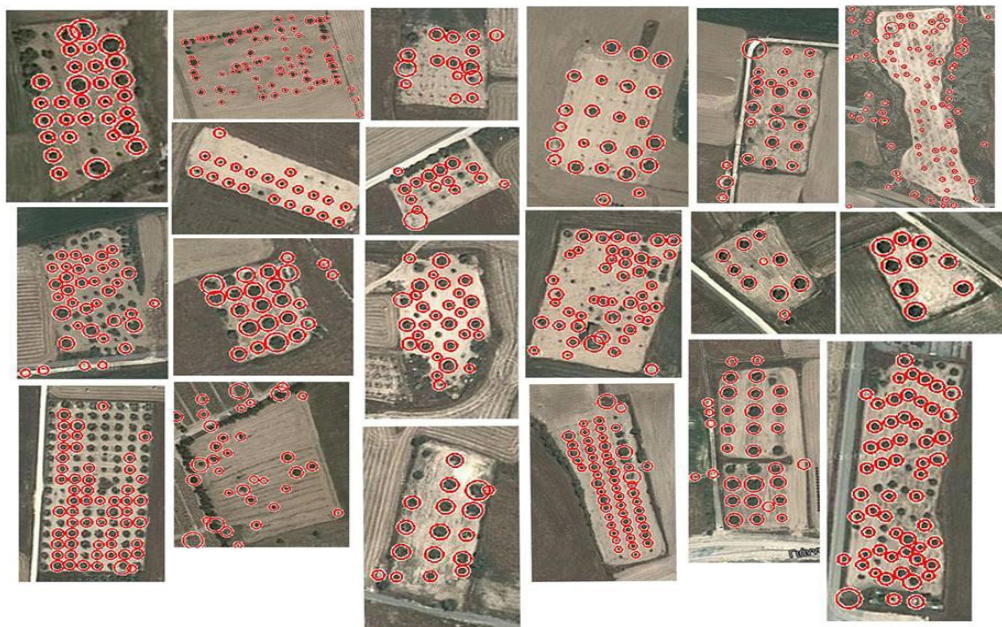
- [15] Kevin Murphy, Antonio Torralba, Daniel Eaton, & William Freeman, (2006). Object detection and localization using local and global features, Toward Category-Level Object Recognition Lecture Notes in Computer Science Volume 4170, pp 382-400
- [16] D. Maheswari, & V.Radha, (2010). Noise removal in compound image using median filter. IJCSE) International Journal on Computer Science and Engineering. Vol. 02, No. 04, 1359-1362
- [17] Marcin Smereka, & Ignacy Duleba, (2008). Circular object detection using a modified Hough transform. Int. J. Appl. Math. Comput. Sci., Vol. 18, No. 1, 85–91
- [18] Jiann-Jone Chen, Chun-Rong Su, W. Eric L. Grimson, Jun-Lin Liu, & De-Hui Shiue, (FEBRUARY 2012). Object Segmentation of Database Images by Dual Multiscale Morphological Reconstructions and Retrieval Applications, IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 21, NO. 2.
- [19] Marie Liénou, Henri Maître, & Mihai Datcu, (JANUARY 2010). Semantic Annotation of Satellite Images Using Latent Dirichlet Allocation, IEEE GEOSCIENCE AND REMOTE SENSING LETTERS, VOL. 7, NO. 1.
- [20] Jacek Kopecky, Tomas Vitvar, Carine Bournez, & Joel Farrell, (2007). SAWSDL: Semantic Annotations for WSDL and XML Schema, : Internet Computing, IEEE Volume:11 , Issue: 6.

## **APPENDIX**

Some samples from canny techniques



Some samples from Sobel techniques



## Matlab code

```
%% Canny technique
clc;
clear all;
warning('off','all');
%% Load image
im=imread('img1/22.jpg');
im1=rgb2gray(im); %Conver color image to grayscale

tic % start timer
BW = edge(im1,'canny'); %finding edges
[centers, radii, metric] = imfindcircles(BW,[4 18]);
disp('Detection');
toc % stop timer

figure,imshow(BW),title('Canny without Smoothing');
centersStrong5 = centers(1:size(centers,1),:);
radiiStrong5 = radii(1:size(centers,1));
metricStrong5 = metric(1:size(centers,1));
viscircles(centersStrong5, radiiStrong5,'EdgeColor','r');
disp('Total');
toc % stop timer

figure,imshow(im),title('Canny without Smoothing');
viscircles(centersStrong5, radiiStrong5,'EdgeColor','r');

total_detections = size(centers,1);
disp('Total detections:');
disp(total_detections);

save('centers_Canny.mat', 'centers') % Save pixel coordinates of trees
save('radii_Canny.mat', 'radii') % save radius of trees

%% Prewitt technique

clc;
clear all;
warning('off','all');
%% Load image
im=imread('img1/15.jpg');
im1=rgb2gray(im); %Conver color image to grayscale

tic % start timer
BW = edge(im1,'prewitt'); %finding edges
[centers, radii, metric] = imfindcircles(BW,[4 18]);
disp('Detection');
toc % stop timer

figure,imshow(BW),title('Prewitt without Smoothing');
centersStrong5 = centers(1:size(centers,1),:);
radiiStrong5 = radii(1:size(centers,1));
metricStrong5 = metric(1:size(centers,1));
viscircles(centersStrong5, radiiStrong5,'EdgeColor','r');
disp('Total');
toc % stop timer

figure,imshow(im),title('Prewitt without Smoothing');
viscircles(centersStrong5, radiiStrong5,'EdgeColor','r');

total_detections = size(centers,1);
disp('Total detections:');
disp(total_detections);

save('centers_Prewitt.mat', 'centers') % Save pixel coordinates of trees
save('radii_Prewitt.mat', 'radii') % save radius of trees

%% Roberts technique
clc;
clear all;
```



```

warning('off','all');
%% Load image
im=imread('img1/15.jpg');
im1=rgb2gray(im); %Conver color image to grayscale

tic % start timer
BW = edge(im1,'roberts'); %finding edges
[centers, radii, metric] = imfindcircles(BW,[4 18]);
disp ('Detection');
toc % stop timer

figure,imshow(BW),title('Roberts without Smoothing');
centersStrong5 = centers(1:size(centers,1),:);
radiiStrong5 = radii(1:size(centers,1));
metricStrong5 = metric(1:size(centers,1));
viscircles(centersStrong5, radiiStrong5,'EdgeColor','r');
disp ('Total');
toc % stop timer

figure,imshow(im),title('Roberts without Smoothing');
viscircles(centersStrong5, radiiStrong5,'EdgeColor','r');

total_detections = size(centers,1);
disp ('Total detections:');
disp (total_detections);

save('centers_Roberts.mat', 'centers') % Save pixel coordinates of trees
save('radii_Roberts.mat', 'radii') % save radius of trees

%% Sobel technique
clc;
clear all;
warning('off','all');
%% Load image
im=imread('img1/37.jpg'); % load image from local storage
im1=rgb2gray(im); %Conver color image to grayscale

tic % start timer
BW = edge(im1,'sobel'); %finding edges
[centers, radii, metric] = imfindcircles(BW,[4 18]);
disp ('Detection');
toc % stop timer

figure,imshow(BW),title('Sobel without Smoothing');
centersStrong5 = centers(1:size(centers,1),:);
radiiStrong5 = radii(1:size(centers,1));
metricStrong5 = metric(1:size(centers,1));
viscircles(centersStrong5, radiiStrong5,'EdgeColor','r');
disp ('Total');
toc % stop timer

figure,imshow(im),title('Sobel without Smoothing');
viscircles(centersStrong5, radiiStrong5,'EdgeColor','r');

total_detections = size(centers,1);
disp ('Total detections:');
disp (total_detections);

save('centers_Sobel.mat', 'centers') % Save pixel coordinates of trees
save('radii_Sobel.mat', 'radii') % save radius

```