Intelligent Vision System

[Intelligent Vision System 2](#_Toc510192228)

[Introduction 2](#_Toc510192229)

[1. Methods and Results 2](#_Toc510192230)

[Fundamental Image Processing 2](#_Toc510192231)

[Static objects segmentation using edge detection 3](#_Toc510192232)

[Basic image processing techniques for decision making 6](#_Toc510192233)

[Background subtraction 7](#_Toc510192234)

[Object motion estimation and tracking 9](#_Toc510192235)

[Find corner points 9](#_Toc510192236)

[Results 13](#_Toc510192237)

[Discussion 14](#_Toc510192238)

[Subsection heading like this 14](#_Toc510192239)

[Conclusions 14](#_Toc510192240)

[References 14](#_Toc510192241)

[Appendix 14](#_Toc510192242)

# Intelligent Vision System

## Introduction

The purpose of this lab is to understand basic image processing methods and developing algorithms such as, decision making, face recognition and tracking. image enhancement, image understanding and video processing for a number of tasks in autonomous systems, including robotics, Unmanned Aerial Vehicles (UAVs), intelligent transportation systems, surveillance, augmented reality and virtual reality systems. This report presents results on reading and writing images in Matlab.

## Methods and Results

The first labs are focused on performing operations on images such as reading, writing finding image histograms, flipping images to extracting the important features colour and edges. You will need to become familiar with how to use these features for the purposes of object segmentation, e.g. for separating static and moving objects and for the higher level tasks of stereo vision, object detection, classification, tracking and behaviour analysis. These are inherent steps of semi-supervised and unsupervised systems where the involvement of the human operators is reduced to minimum or excluded.

### Fundamental Image Processing

In this section basic operations on images of different types are implemented, such as image histograms and visualisation.

It clearly indicates that the two-colour image contains more colour bins than one-color image. It is possible to decide the image’s characters according to image histogram.

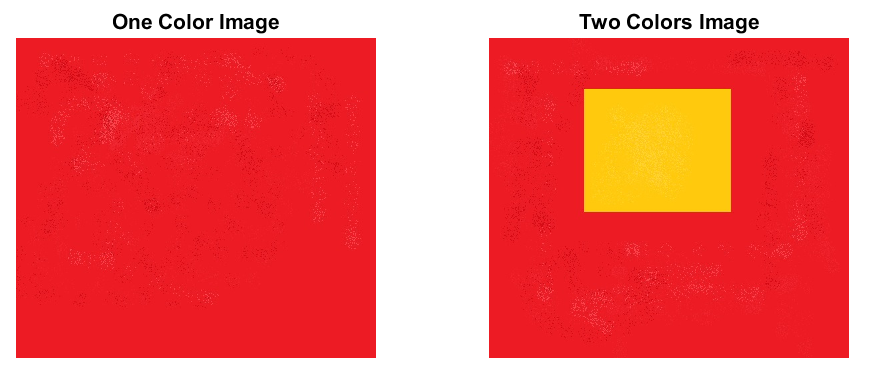
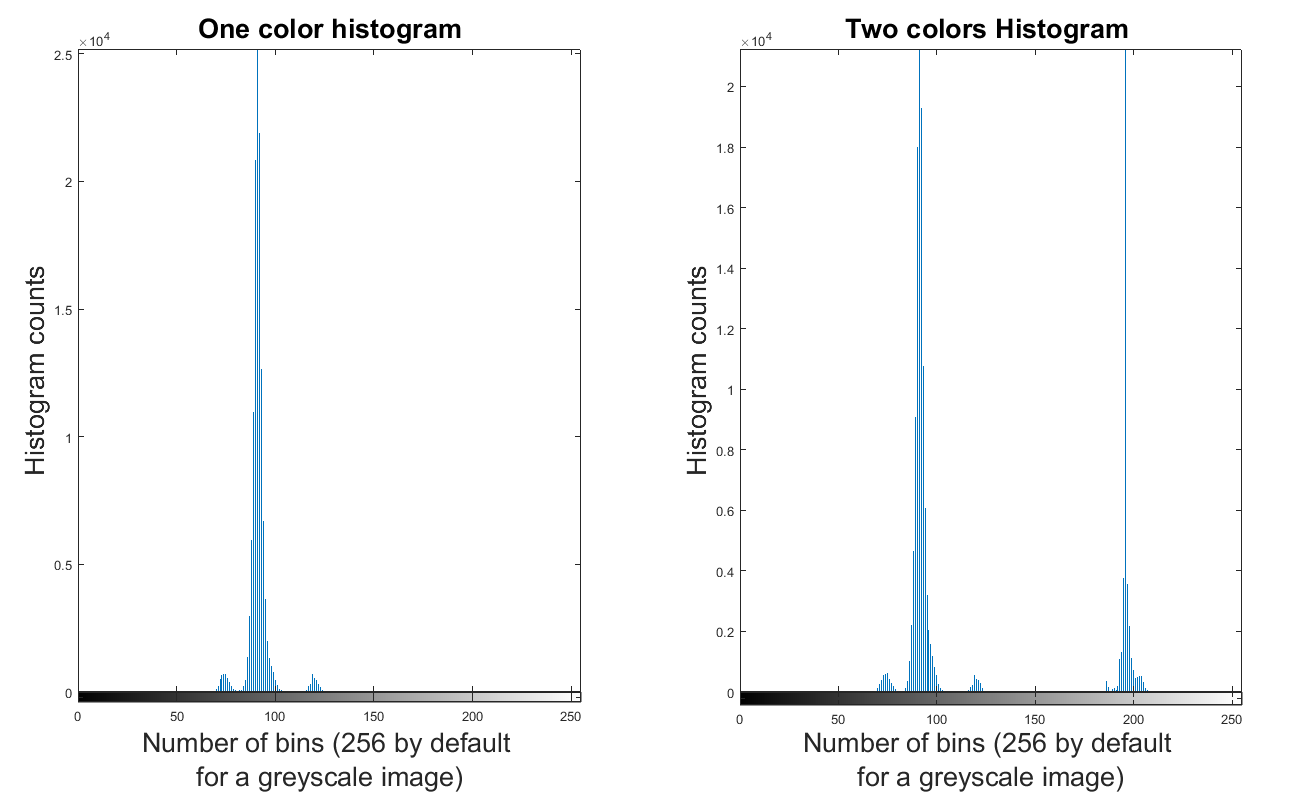


Figure 1 Two images



**Figure 2 Histogram Comparison between one-color image and two-colour image**

After converting the two images to gray format, figure 2 indicates that histogram of one color image’s values are mainly distributed around 100. But on the right side of figure 2, It clearly shows that the two-colour image contains some colour bins which are located around the value 200. It is possible to decide the image’s characters according to image histogram. As a result, it can be straightforward to distinguished the two images according to the histograms.

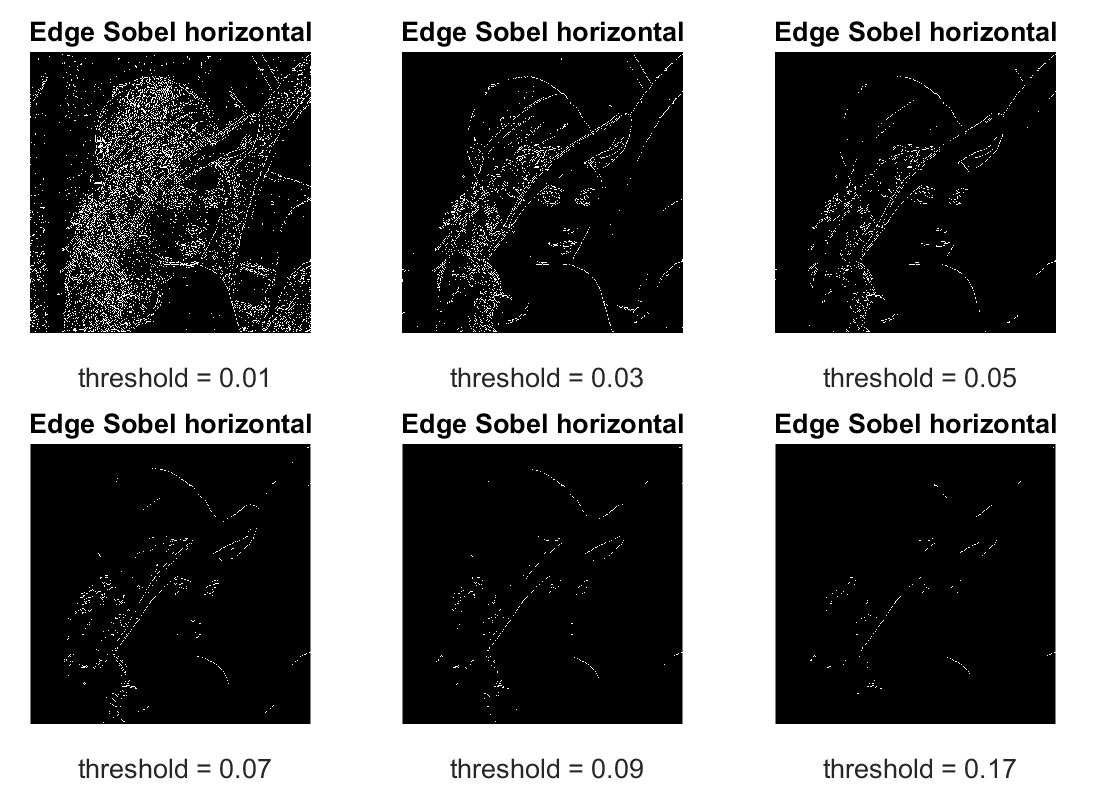
### Static objects segmentation using edge detection

The basic image processing technology will be applied in this part, for example, enhancing the contrast of the image, modelling different types of noise in the image, removing noise from the image, edge detecting and static object segmentation.

There are several methods for static objects segmentation using edge detection such as Sobel, Canny and Prewill.

Sobel operator: Because of the weighted smoothing of the image, it has a certain ability to suppress the noise. But it cannot completely rule out the false edges that appear in the detection result. On the other hand, there exist some disadvantages of the Sobel method that it is sensitive to the noise. That is, it is sensitive to the noise. The magnitude of the edges will degrade as the level of noise present in image increases [1].

Canny operator: This operator uses a Gaussian function to smooth the image, so it has a strong ability to suppress noise. At the same time, it uses dual threshold detection and connection edges, and is better in multi-scale detection and directional search. The main disadvantage of Canny edge detector is that it is time consuming, due to its complex computation.



**Figure 3 Edge detection with function sable**

Edge detection with function canny.

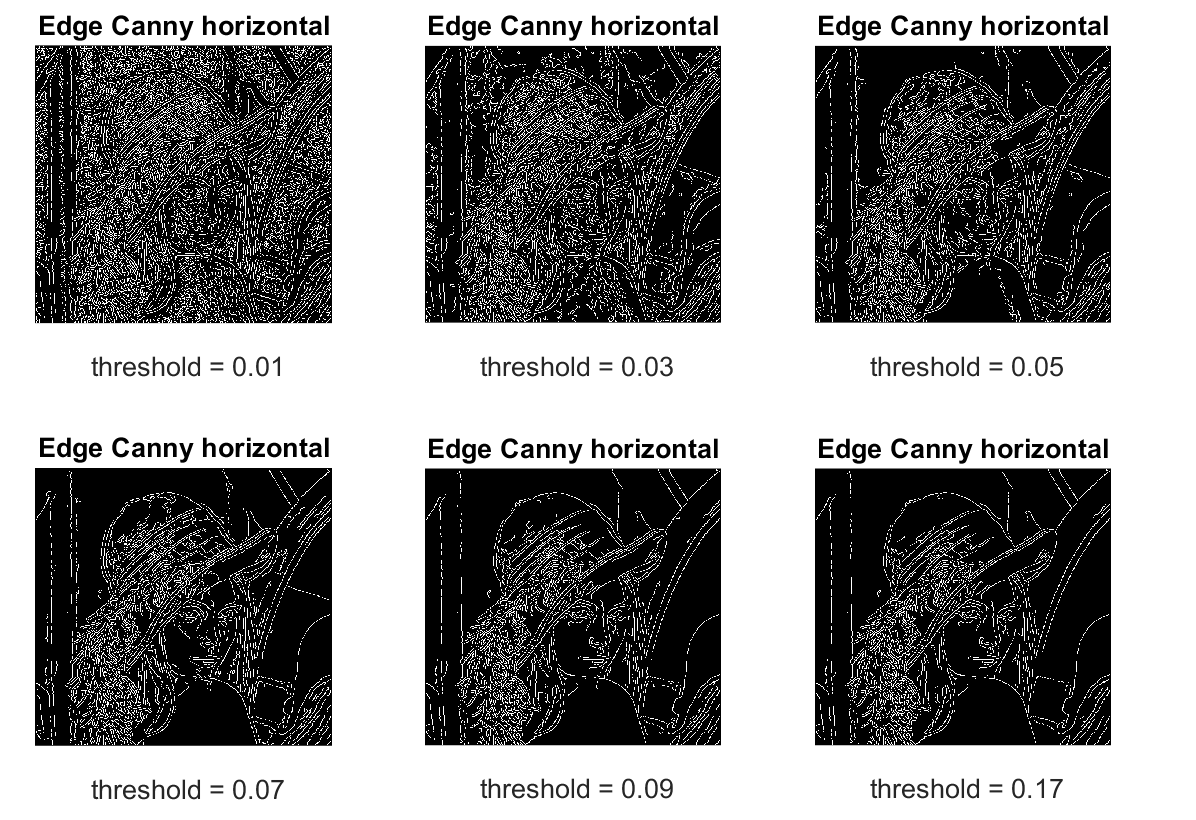
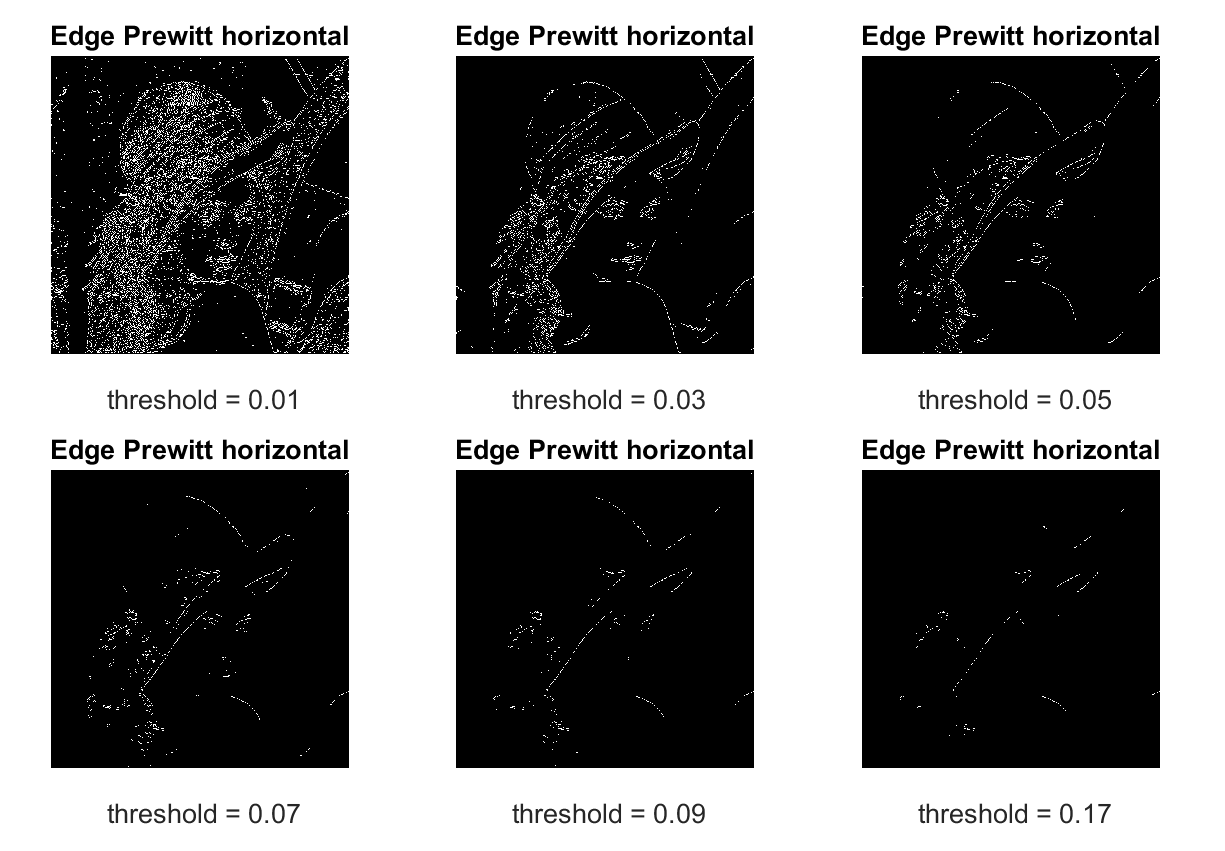


Figure 4 Edge detection with function canny.

Edge detection with function prewitt.



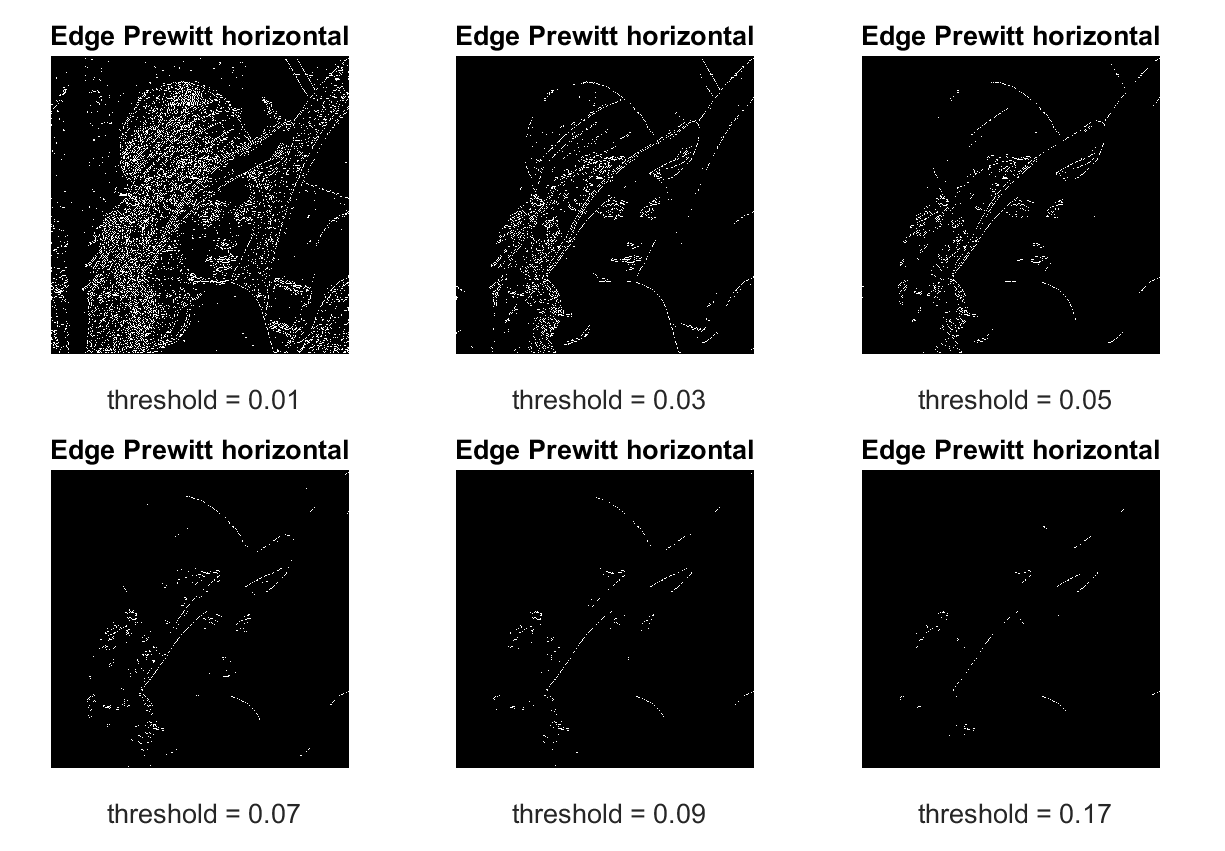


Figure 5 Prewitt

|  |  |  |
| --- | --- | --- |
| Operator | Strengths | Weaknesses |
| Sobel | Simple. Detects edges and their orientation | Inaccurate and sensitive to noise |
| Canny | Smoothing effect to remove noise. Good localization and response. Enhances signal to noise ratio. Immune to noisy environment. | Difficult to implement to reach real time response. Time consuming. |
| Prewitt |  |  |

Table 2 Compares strengths and weaknesses of detectors

### Basic image processing techniques for decision making

Convert all given images in binary format

IM2BW produces binary images from indexed, intensity, or RGB images. To do this, it converts the input image to grayscale format (if it is not already an intensity image), and then converts this grayscale image to binary by thresholding. The output binary image BW has values of 1 (white) for all pixels in the input image with luminance greater than LEVEL and 0 (black) for all other pixels. (Note that you specify LEVEL in the range [0,1], regardless of the class of the input image.

The input image can be uint8, uint16, single, int16, or double and it must be no sparse. The output image BW is logical. I and X must be 2-D.RGB images are M-by-N-by-3.

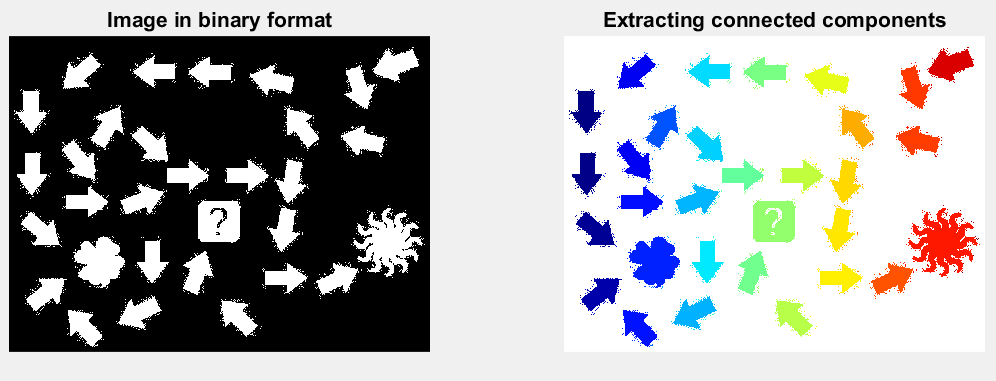


Figure 6 Threshold = 0.01

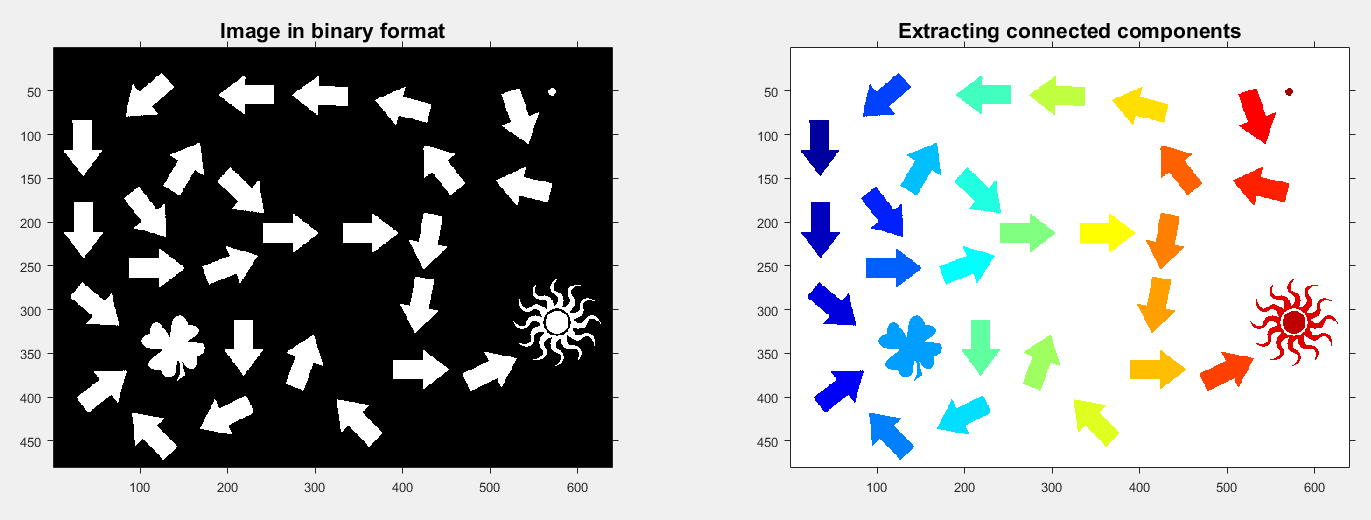


Figure 7 Threshold = 0.5

Function arrow\_finder -----Arrow/Non-arrow detection

As it is done in last part, bounding box were founded by using the function **bwlabel** and **regonprops.** Function **regionprops** is used to measure the properties of the specified area. After implementing this function, variable **props** is generated. Its data type is structure with three fields, Area, Centroid and BoundingBox. In filed, it will return a scalar that specifies the actual number of pixels in the region. According to this, it can be summarised that the all the arrows have the identified character which is less than 1700.This value is used to distinguish arrow and non-arrow objects in function arrow\_finder.m.

Function next\_object\_finder

After finding the red arrow, treasure finding path is initialised. To find the treasure in image, following steps are implemented for setting up the algorithm.

First of all, the central point of bounding box and central point of yellow area in each bounding box are extracted.

Secondly, one linear equation can be set up according to those two points for each bounding box. Arrow’s direction can be identified by a vector (x2-x1, y2-y1), which (x2, y2) represents central yellow point in the bounding box and (x1, y1) represents central point of the bounding box respectively.

Thirdly, the next object searching can be initialized according to the linear equation. Column will be increased or decreased according to the vector direction. At the same time, the pixel values of three consecutive points with a space of one are used as the basis for finding the next object. When the value of those three pixels for all channels When those three points’ pixel values are meet the corresponding threshold range, it indicated that the next target object has been found.

当三个像素点的值依次满足在相应的阀值范围时, 此时我们判断下一个目标物体已经找到。

Lastly, the variable cur\_object will be assigned to main programme until the treasure was found.

### Background subtraction

Background subtraction, also known as foreground detection, is a technique in the fields of image processing and computer vision wherein an image's foreground is extracted for further processing. Generally, objects such as human, cars, text in foreground are the interesting region in image processing and detecting. In this section two main methods of background subtraction will be compared with different threshold.

**Motion detection base on Frame differencing algorithm**

Due to its low computational load, frame differencing algorithm is the simplest form of background subtraction. Moving objects are detected by the pixel-based values ‘changing.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Threshold** | **Tree** | **Passage** | **Cars** |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |



Figure 8 Frame Difference

Gaussian mixture model algorithm for background subtraction

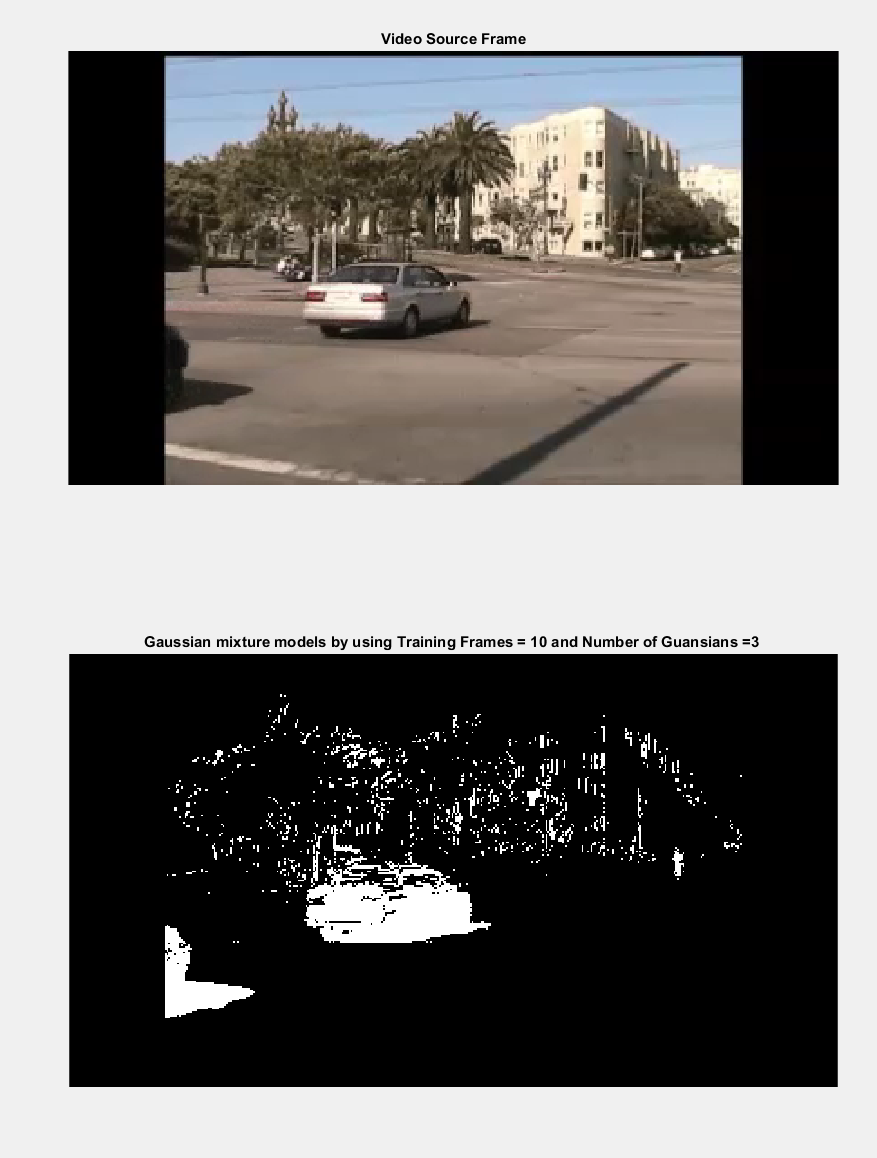


Figure 9 Gaussian mixture model algorithm for background subtraction

### Object motion estimation and tracking

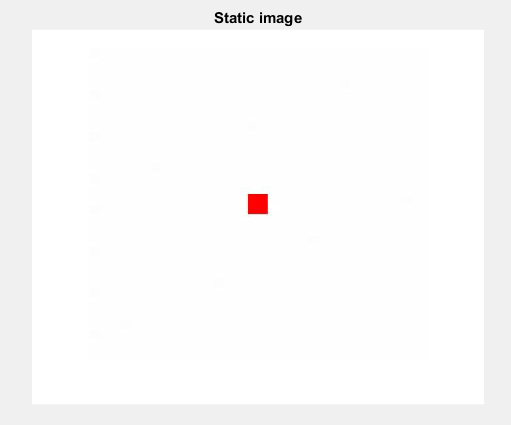
Normal text h

### Find corner points

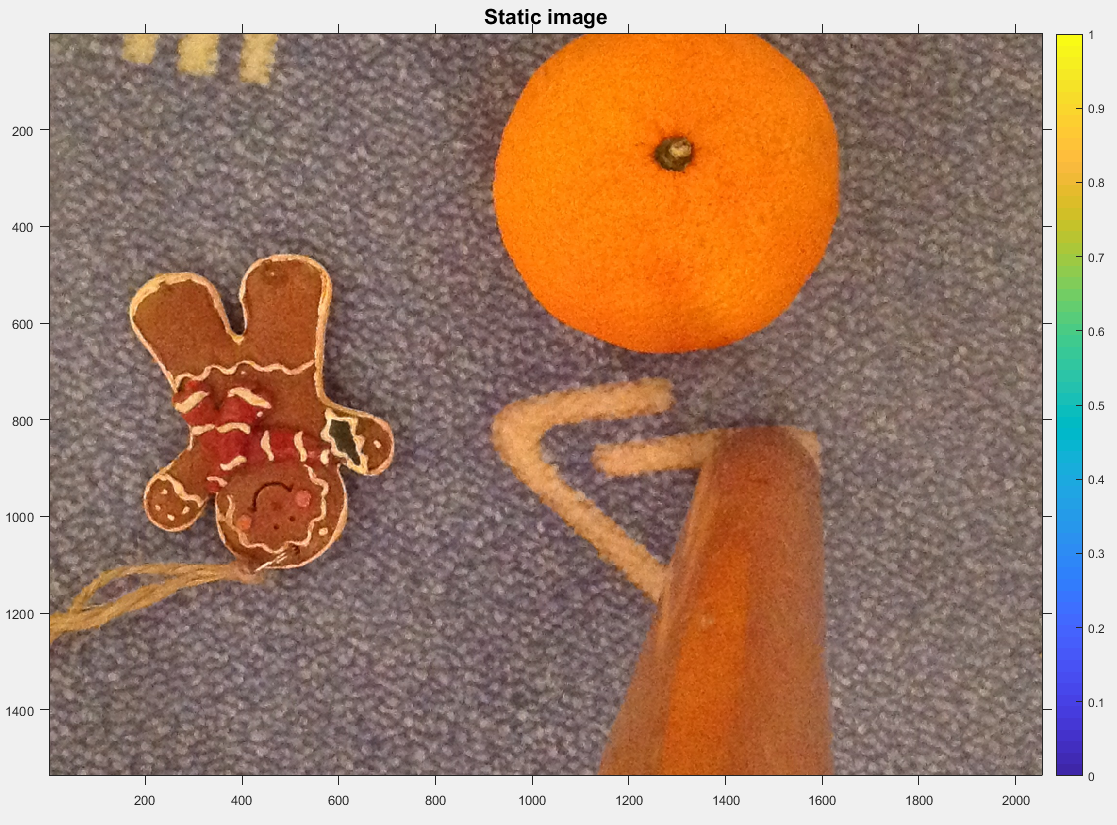
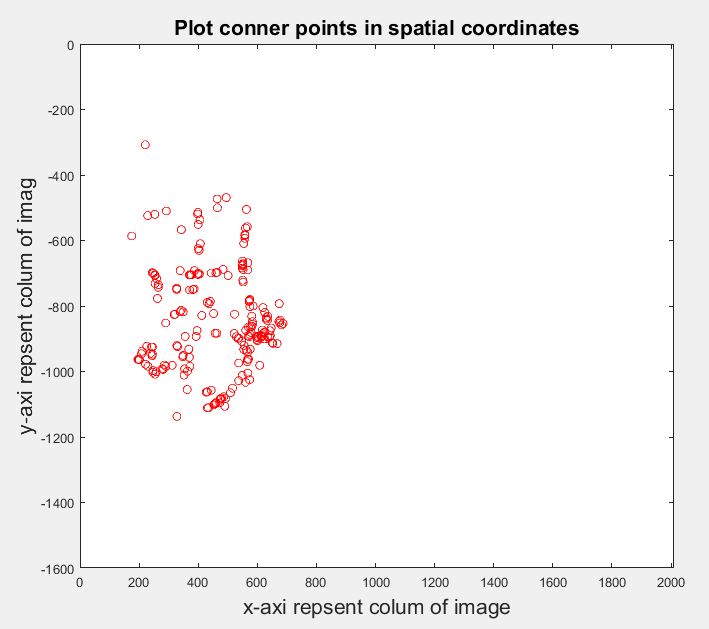
Find corner point with function corner.

[C](file:///C:\Program%20Files\MATLAB\R2017a\help\images\ref\corner.html?browser=F1help#outputarg_C?browser=F1help)orner\_point = corner([I](file:///C:\Program%20Files\MATLAB\R2017a\help\images\ref\corner.html?browser=F1help#inputarg_I?browser=F1help)mage) detects corners in image and returns them in matrix Corner\_point.

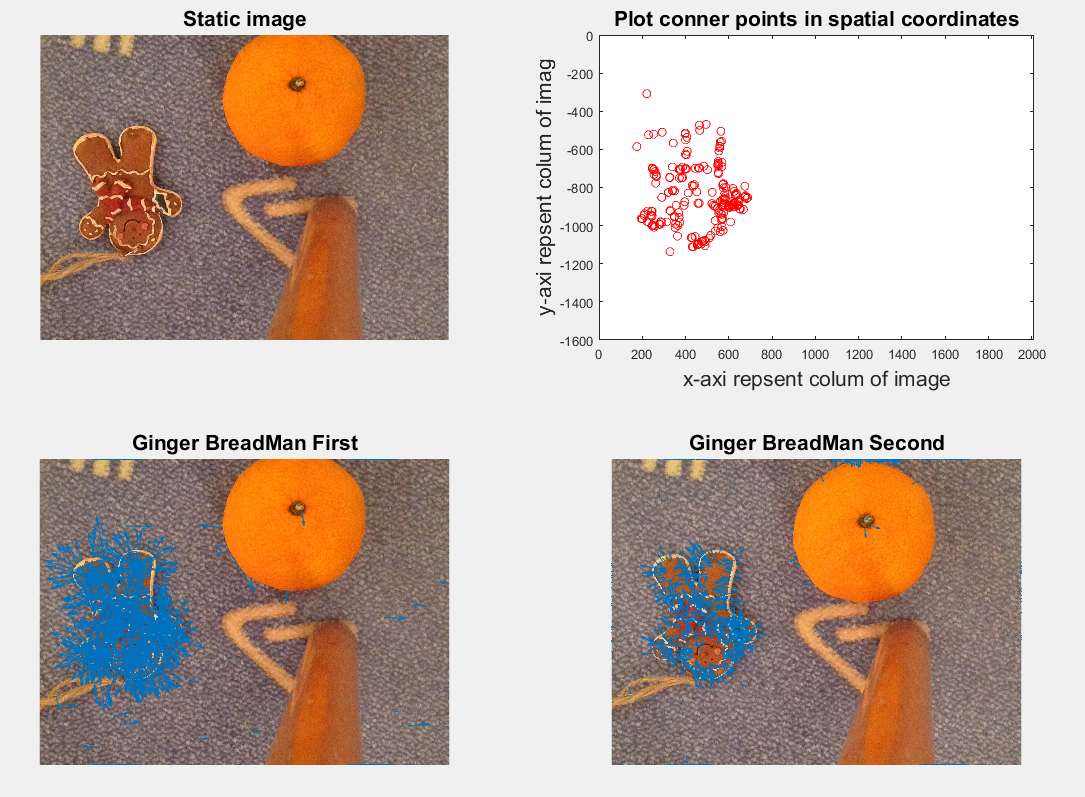
After implemented function ***corner*** in Matlab, four points are found for image red\_square\_static.jpg.

With image GingerBreadMan\_first.jpg, there are 200 corner points were found.

Optical flow of the pixels finding.



Visualise the track on the last frame of the video

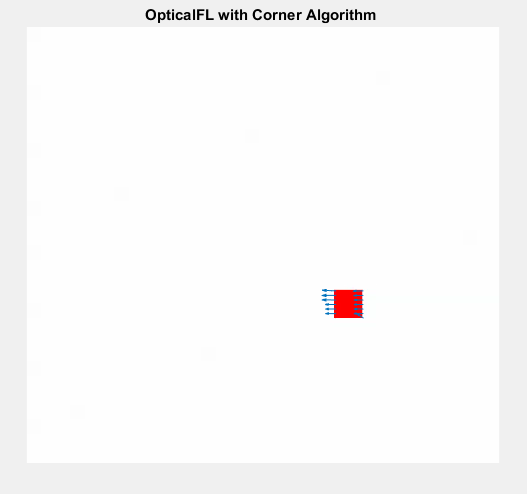


Figure 10 Last frame for track

estimated track by the optical flow algorithm in comparison with the ground truth values

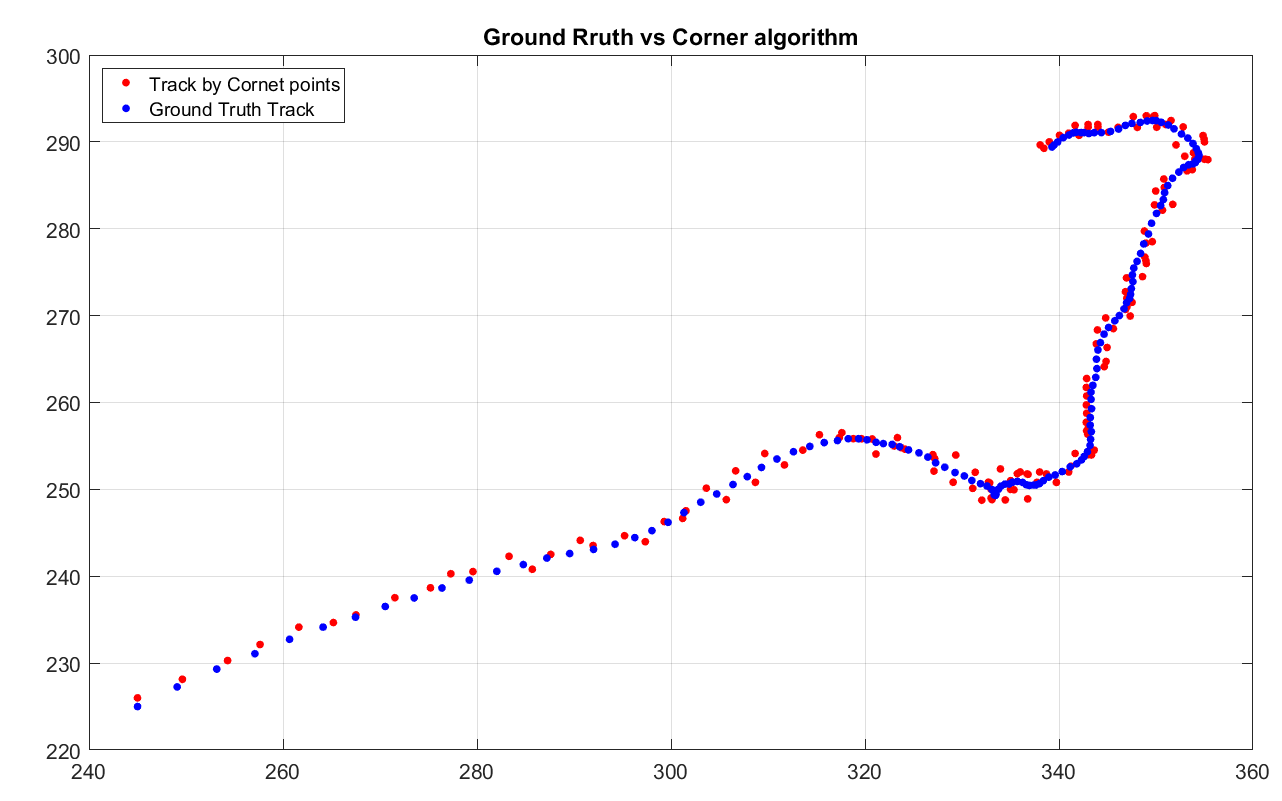
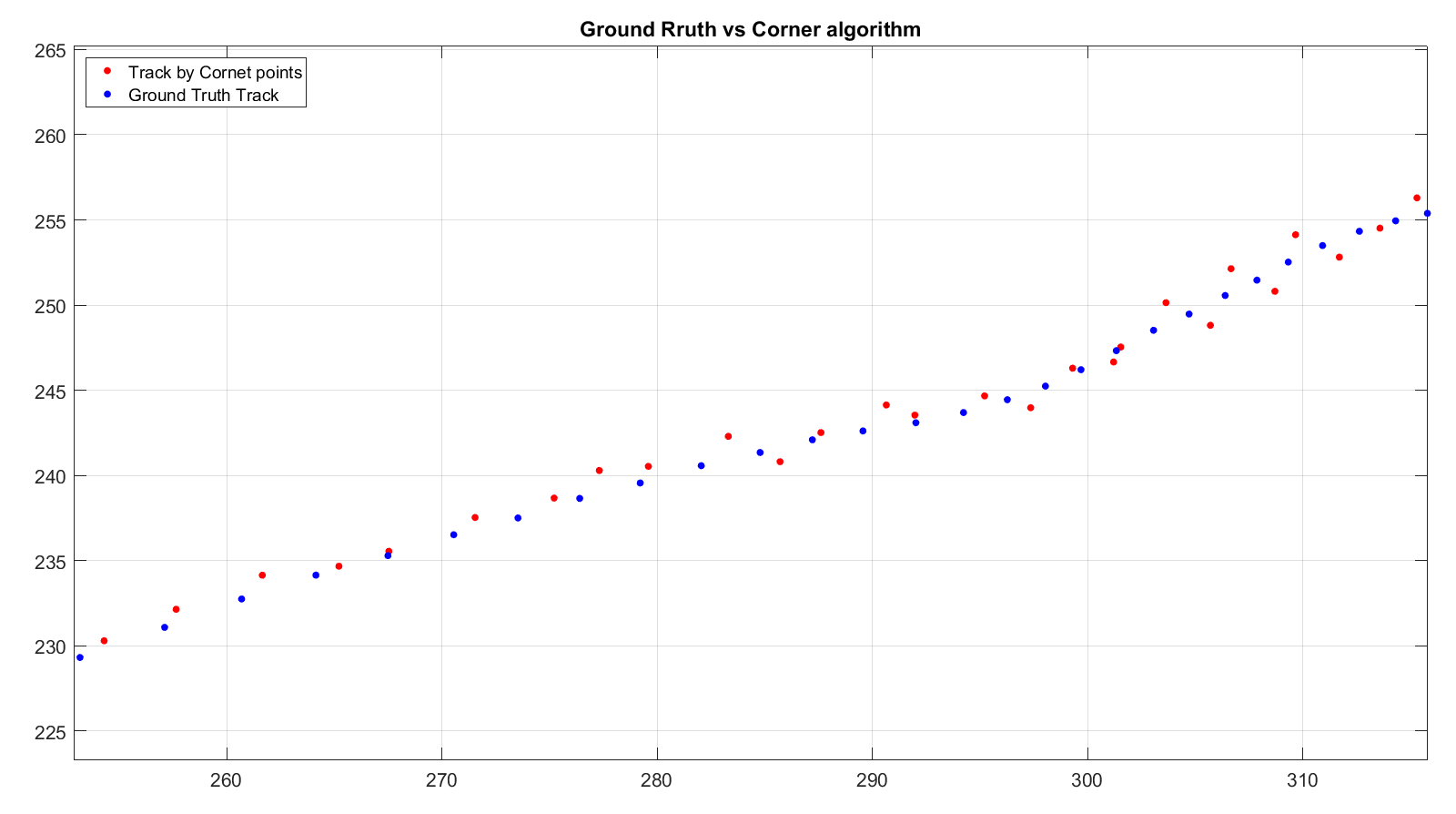


Figure 11 Truth and track points compilation



Face detection and tracking

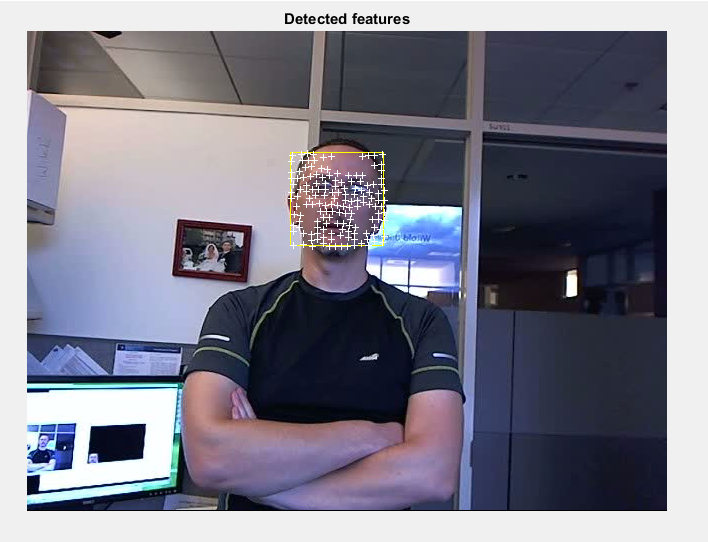
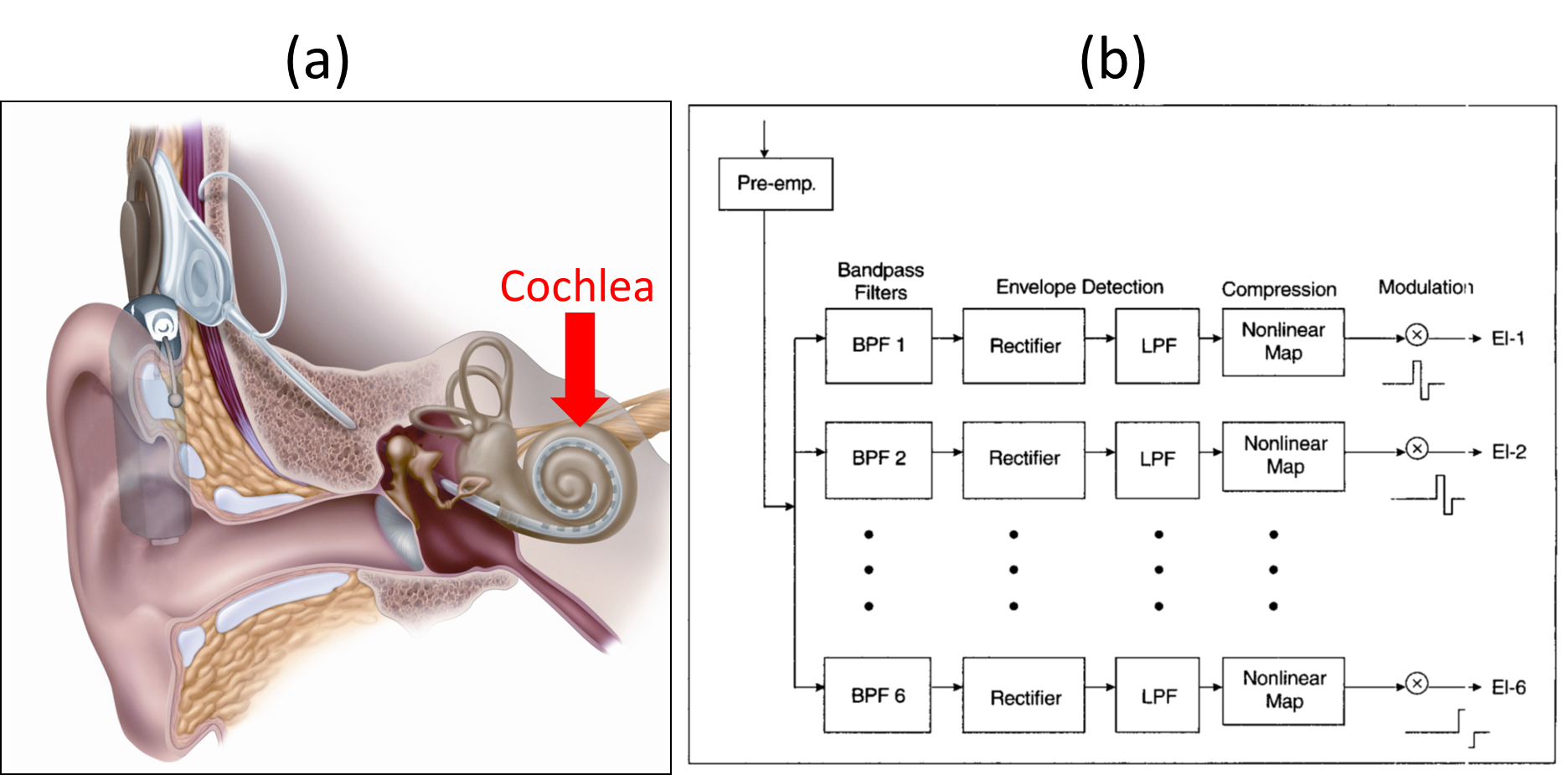


Figure 12 Face detection and tracking

Figure like this (include labels and caption):

**Figure 13** (a) A cochlear implant (Zeng, et al. 2008). (b) Signal processing block diagram for the continuous interleaved sampling procedure (Laizou 1999). Note: do not actually copy/paste low quality figures like this one in (b) from journal papers (image is fuzzy and text is so small it is almost unreadable) – redraw the figure instead.

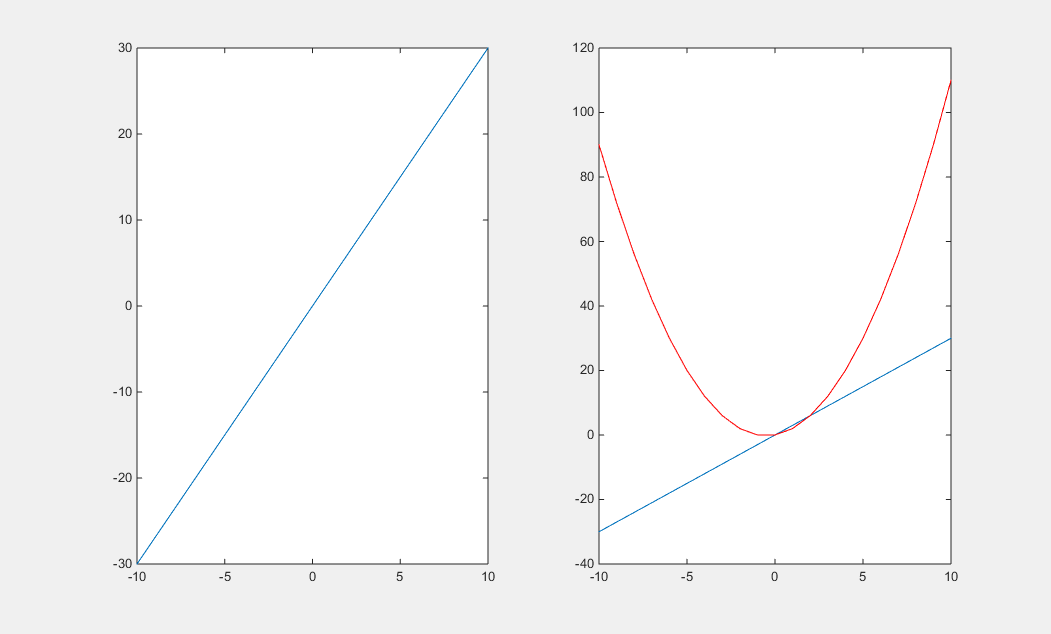
## Results

Normal text here.

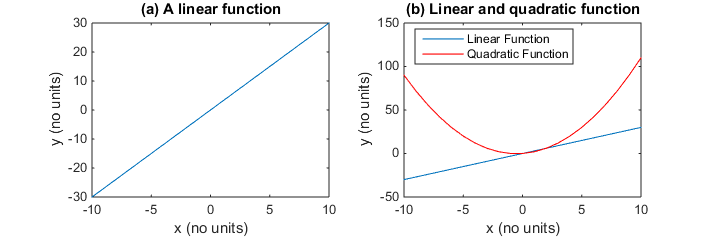
Please do the following with figures:

1. Text should be a readable size (hint: shrink your Matlab figure before exporting it, which makes the text relatively larger).
2. Label axes.
3. Include units or state e.g. no units or arbitrary units (a. u.)
4. If plotting more than one line include a legend.
5. Include labels and titles if appropriate (use your judgement).
6. Figures do not have to take up a lot of space – size them appropriately.

Bad figure:



Good figure:



**Figure 14** (a) A linear function, y=3x, demonstrating linearity! (b) A comparison of a linear and quadratic function, demonstrating nonlinearity!

## Discussion

Normal text here

### Subsection heading like this

## Conclusions

Conclusions here.

Note: References can be automatically inserted below by going to the References menu tab in MS Word and clicking ‘Bibliography’->’Insert Bibliography’.

### References

[1] D. Kim, “Sobel Operator and Canny Edge Detector ECE 480 Fall 2013 Team 4,” pp. 1–10, 2013.

Laizou, P. C. 1999. “Signal-processing techniques for cochlear implants.” *IEEE Engineering in Medicine and Biology Magazine* 18 (3): 34-46.

Zeng, F. G., S. Rebscher, W. Harrison, X. Sun, and H. Feng. 2008. “Cochlear implants: system design, integration, and evaluation.” *IEEE Reviews in Biomedical Engineering* 1: 115-142.

## Appendix

Normal text here