

# Introduction to Computer Vision

## Image Corner Detection Using Harris Corner Detection

-Yashwanth Telekula

### Summary:

We are trying to find the corner points of an image by using the method of Harris corner detection. To find the corners of an image with maximum accuracy, we follow a series of steps that are mentioned in the assignment. The series of steps are explained as follows.

- In order to run the code, run the maincode.m file and manipulate the image reading lines in the maincode.m file to test the project on different images.

### Code Explanation:

- Maincode.m
- detectHarrisCorners.m
- imagesmooth.m
- maxvalues.m
- erase.m

#### 1. Maincode:

This is the main function that has to be run to get the corner detected image output. In this file we read the image, assign the S,N,D,M values and call the detectHarrisCorners function. The values of S,N,D,M and image can be changed in this file to get variable results.

#### 2. DetectHarrisCorners:

This function takes the S,N,D,M and image(2-D matrix) values as inputs and returns the 'corners' and 'R' value matrix. This is the key function. This function calls a lot of functions like imagesmooth, findgradientimage etc. in order to smooth the image, find the gradient image, compute Harris corner 'R' values etc.

```
function [corners, R] = detectHarrisCorners(image, S, N ,D, M)
```

#### 3. image\_smooth:

This function takes the 2-D matrix image and S(sigma) value as the inputs and smoothes the image by calculating the Gaussian matrix and returns the smoothed image.

```
function [ image ] = imagesmooth( image, S )
```

4. **Maxvalues:**

This function takes the R value matrix and tries to find the M maximum value points from the matrix. For every iteration after finding the max value, this function calls the erase function.

```
function [ corners ] = maxvalues( R,D,M )
```

5. **Erase:**

This function takes a co-ordinate value and makes all the values around D-radius region around it as minus infinity.

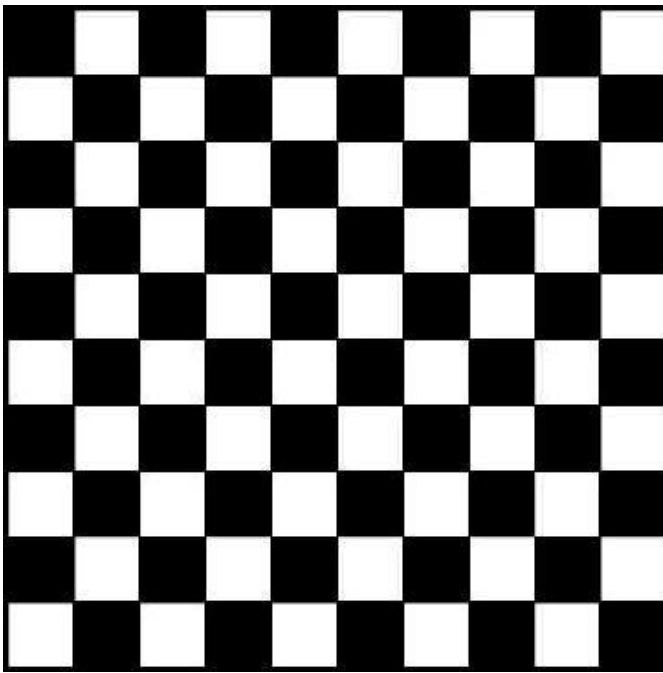
```
function [ R ] = erase( R,x,y,D )
```

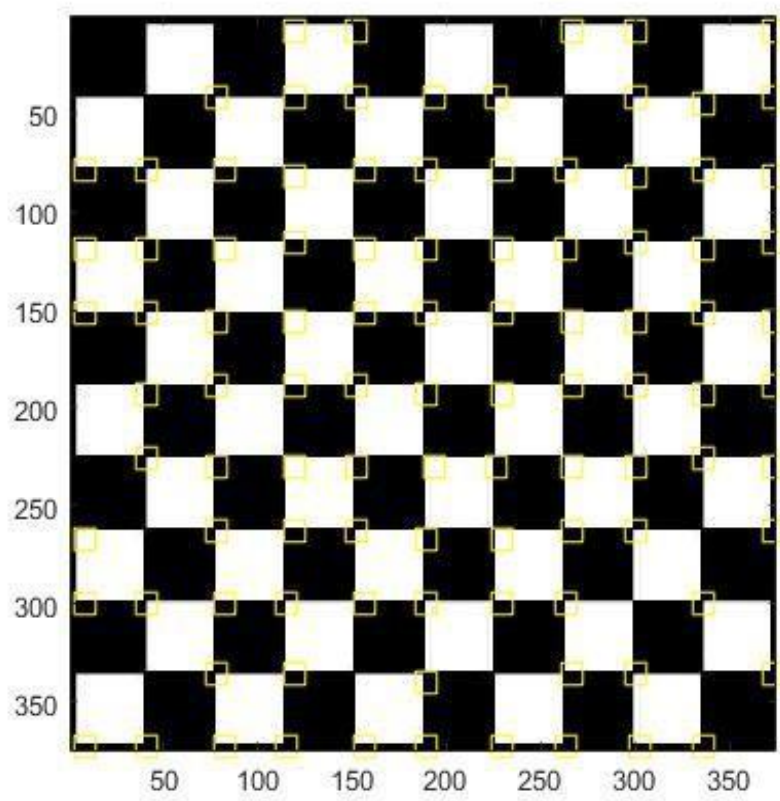
### **Eigenvalue usage:**

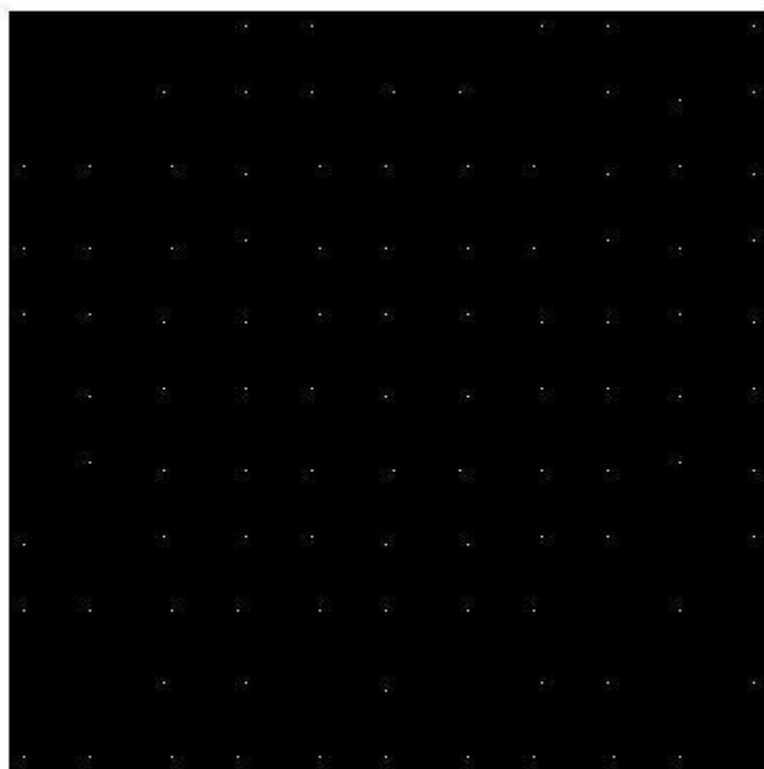
If we use the eigenvalue function to find the highest M points, some of those points maybe close to each other and removing them becomes more hectic making the program more complex. So using the Harris corner detection is a better option.

### **Project Outputs:**

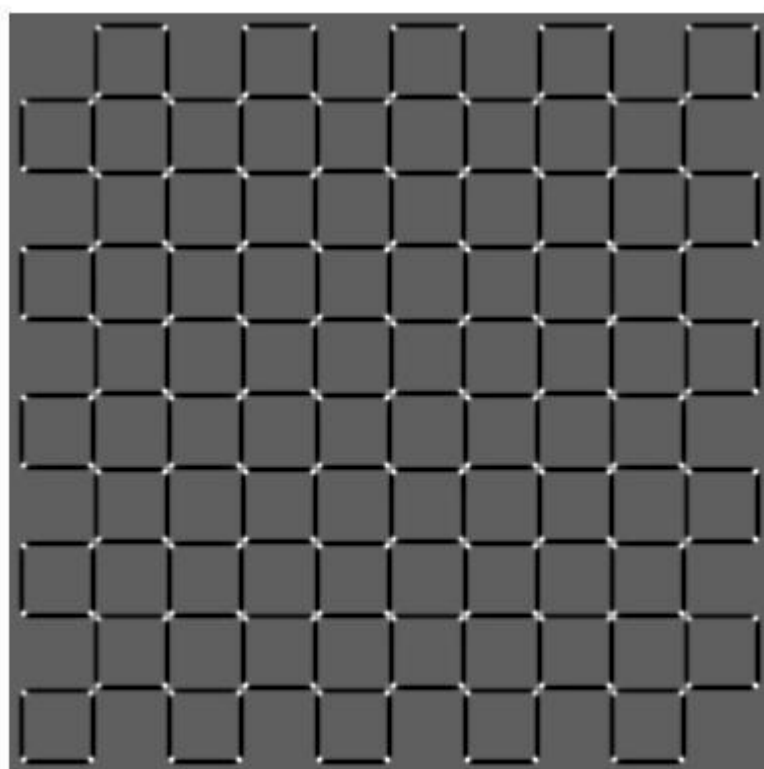
Original image, output image, corner point image and R value image respectively.



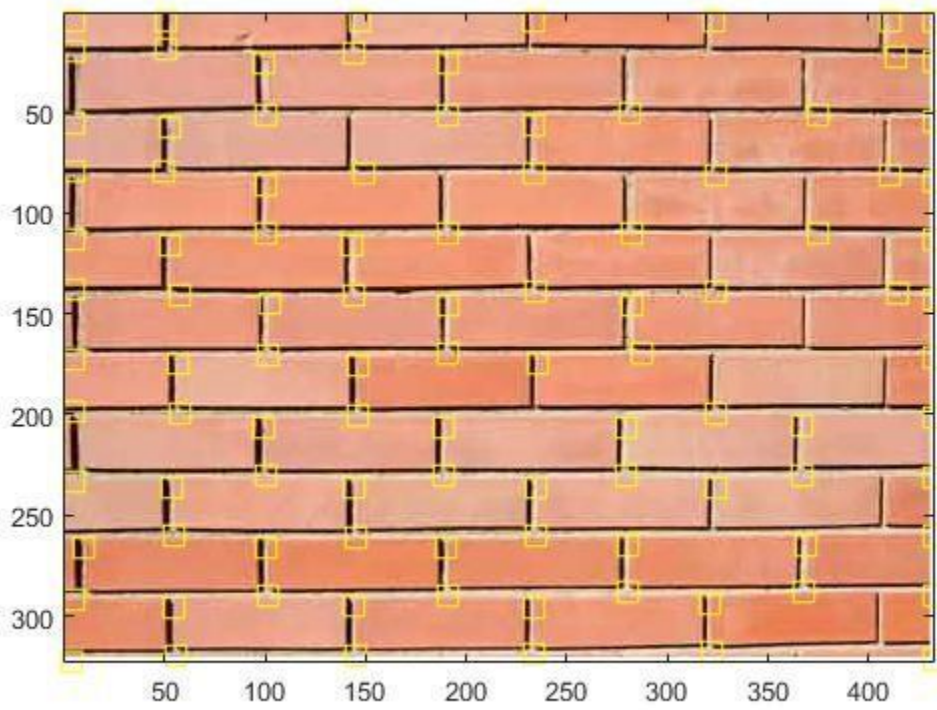
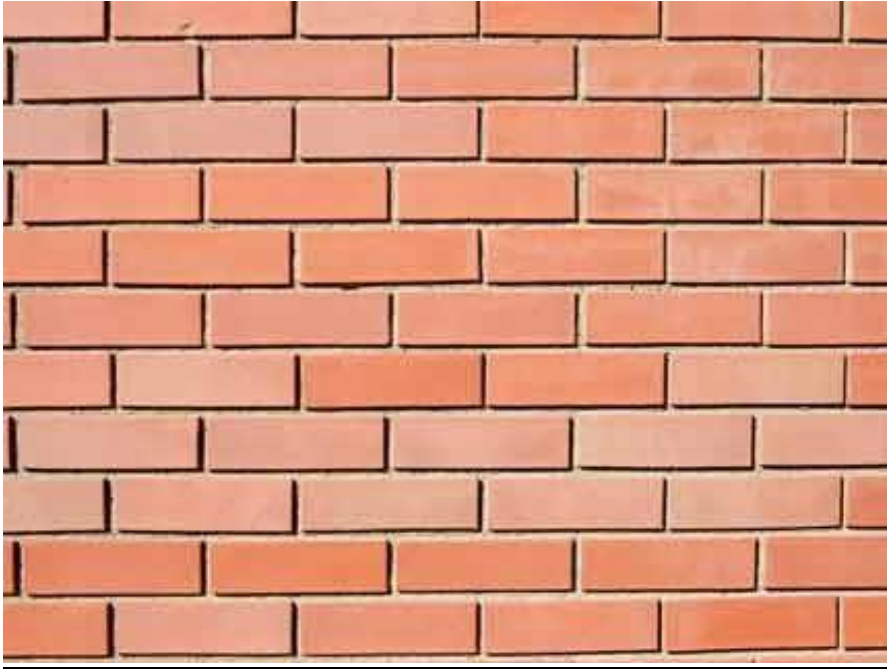


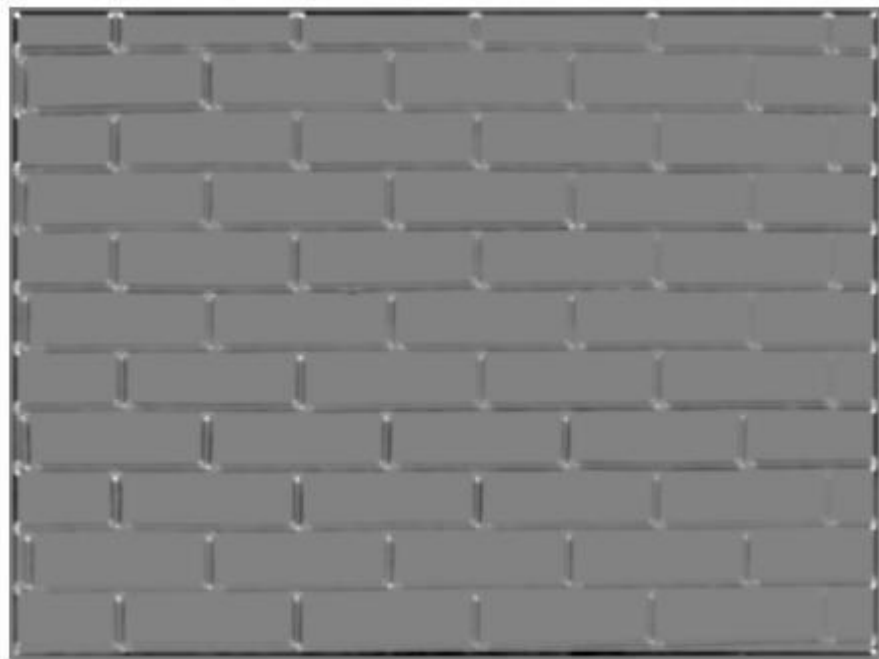
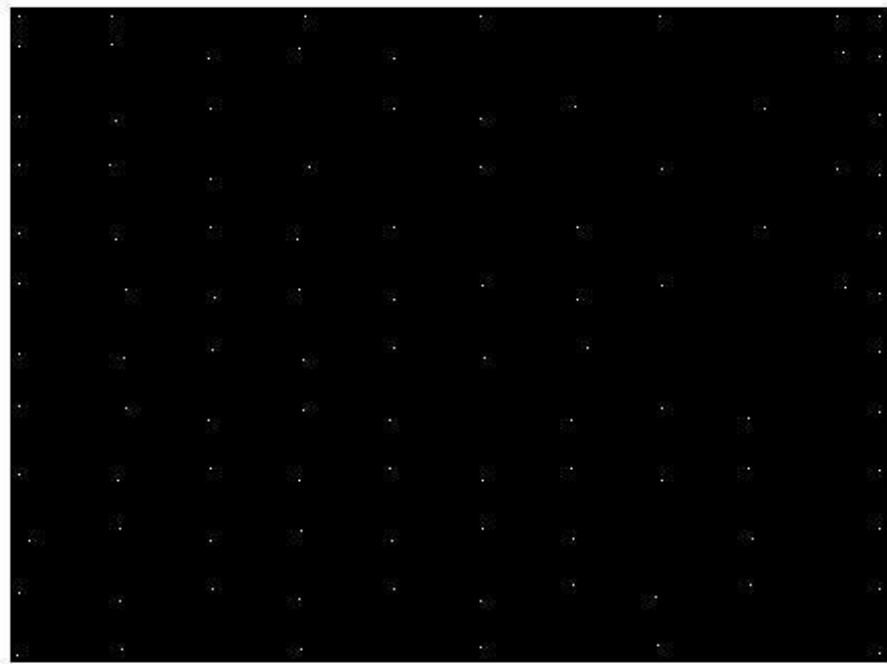


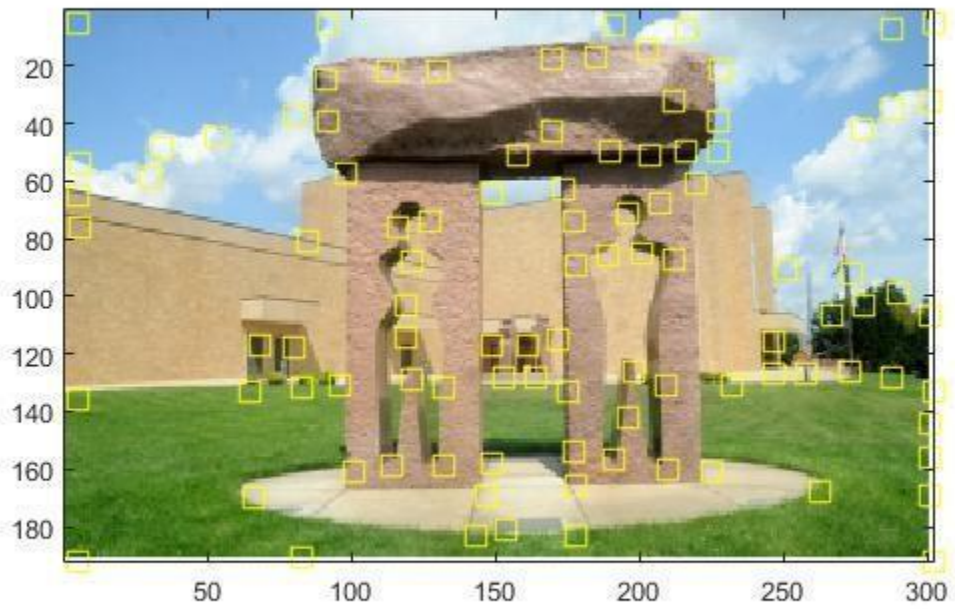
---



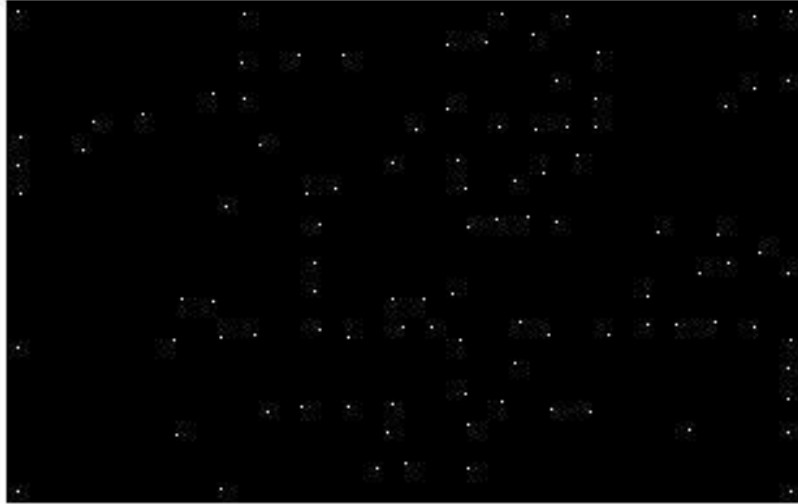
---



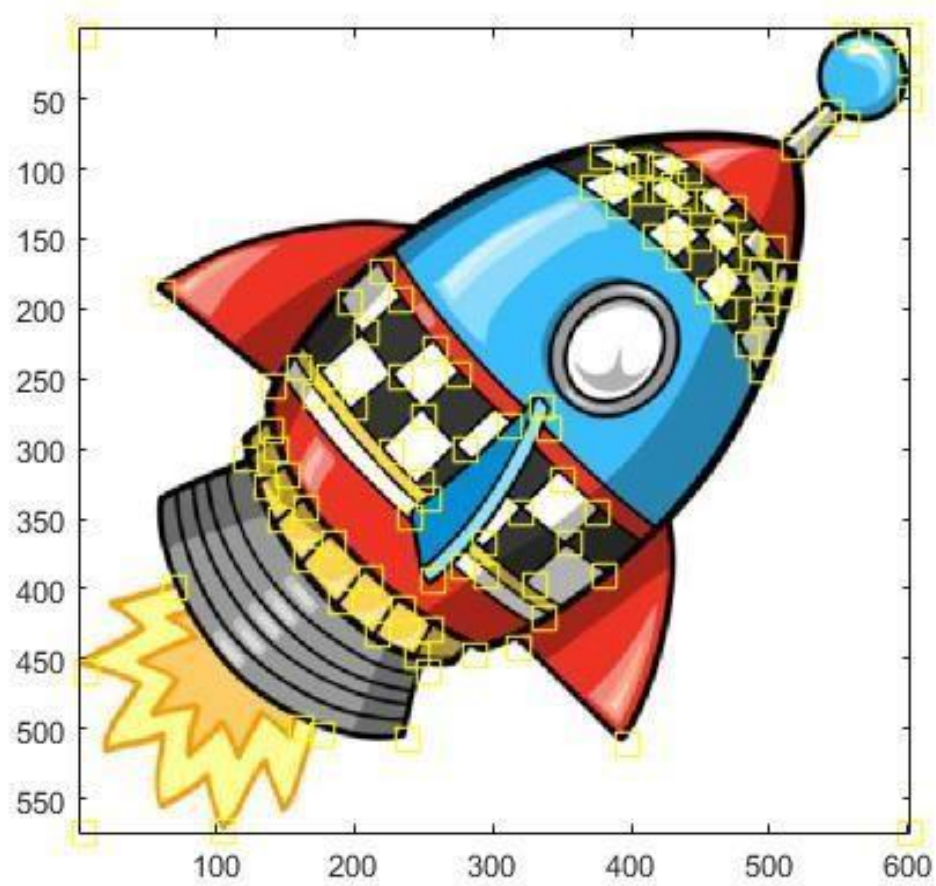


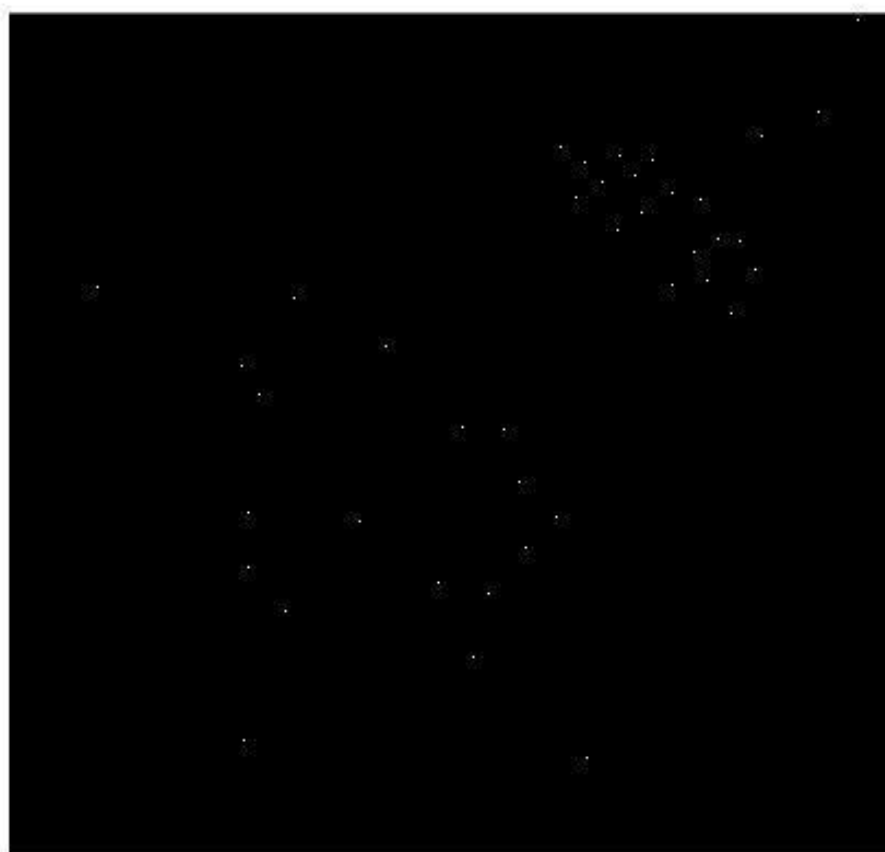




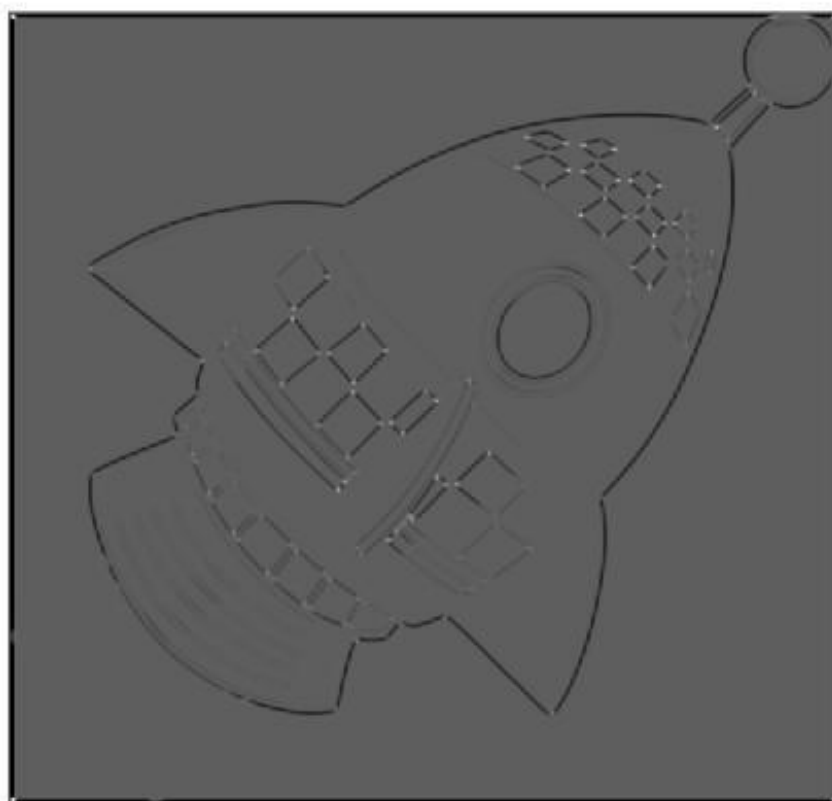








---



---