

## Assignment 2

Original Image1: Image 3\_1.jpg



No. of Octaves - 3

No. of Layers - 5

(a) Scale-space decomposition

(i) Gaussian scale-space images

1. Octave 1

First Layer :



Second Layer :



Third Layer :



Fourth Layer :



Fifth Layer :



## 2. Octave 2

First Layer :

Gaussian scale-space decomposition of Second Octave



Second Layer :





Third Layer :



Fourth Layer :



Fifth Layer :



### 3. Octave 3

First Layer :



Second Layer :



Third Layer :



Fourth Layer :



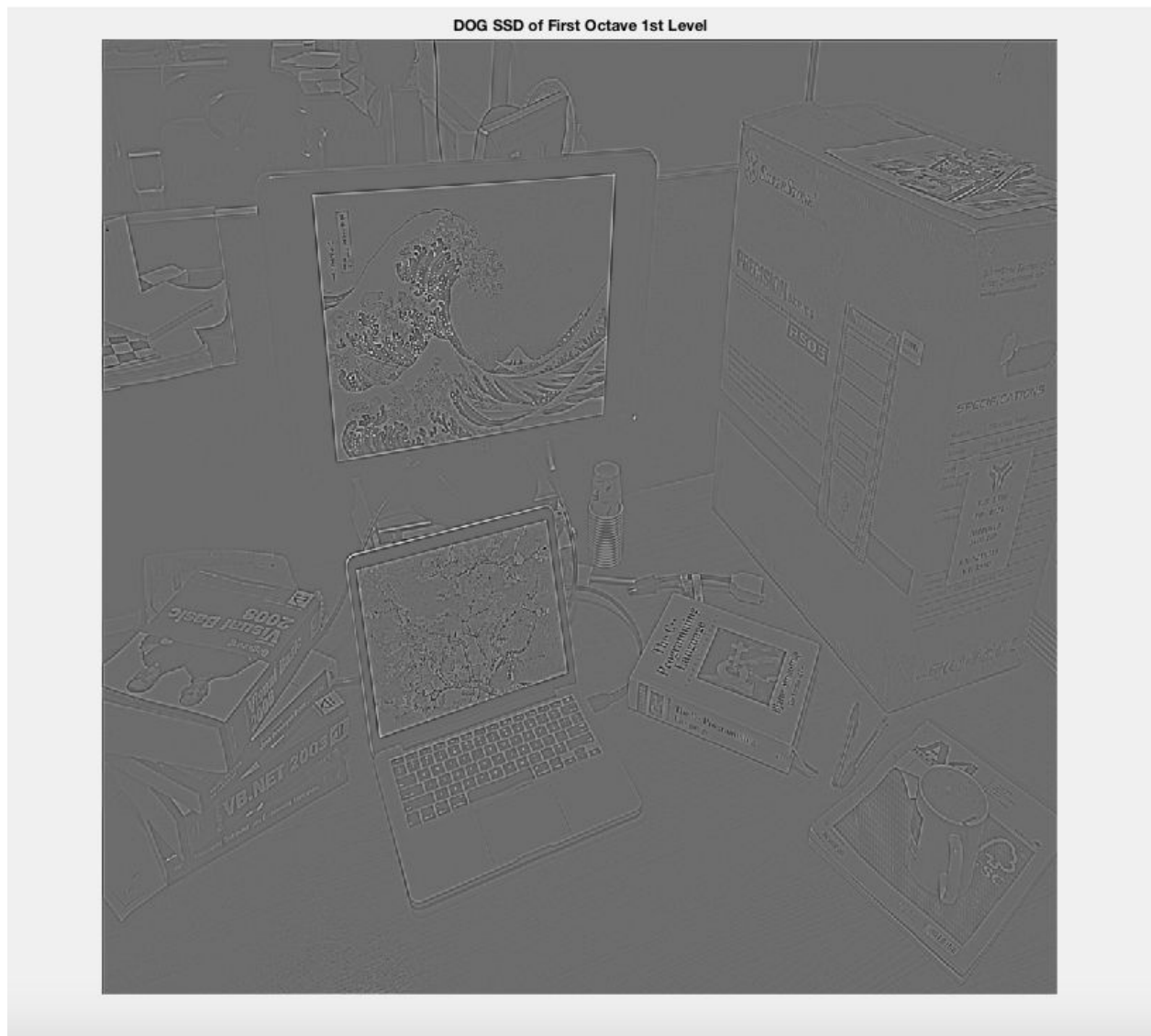
Fifth Layer :

Gaussian scale-space decomposition of Third Octave



- ii) DOG scale-space images
  - 1. Octave 1

First Level :



Second Level :





Third Level :

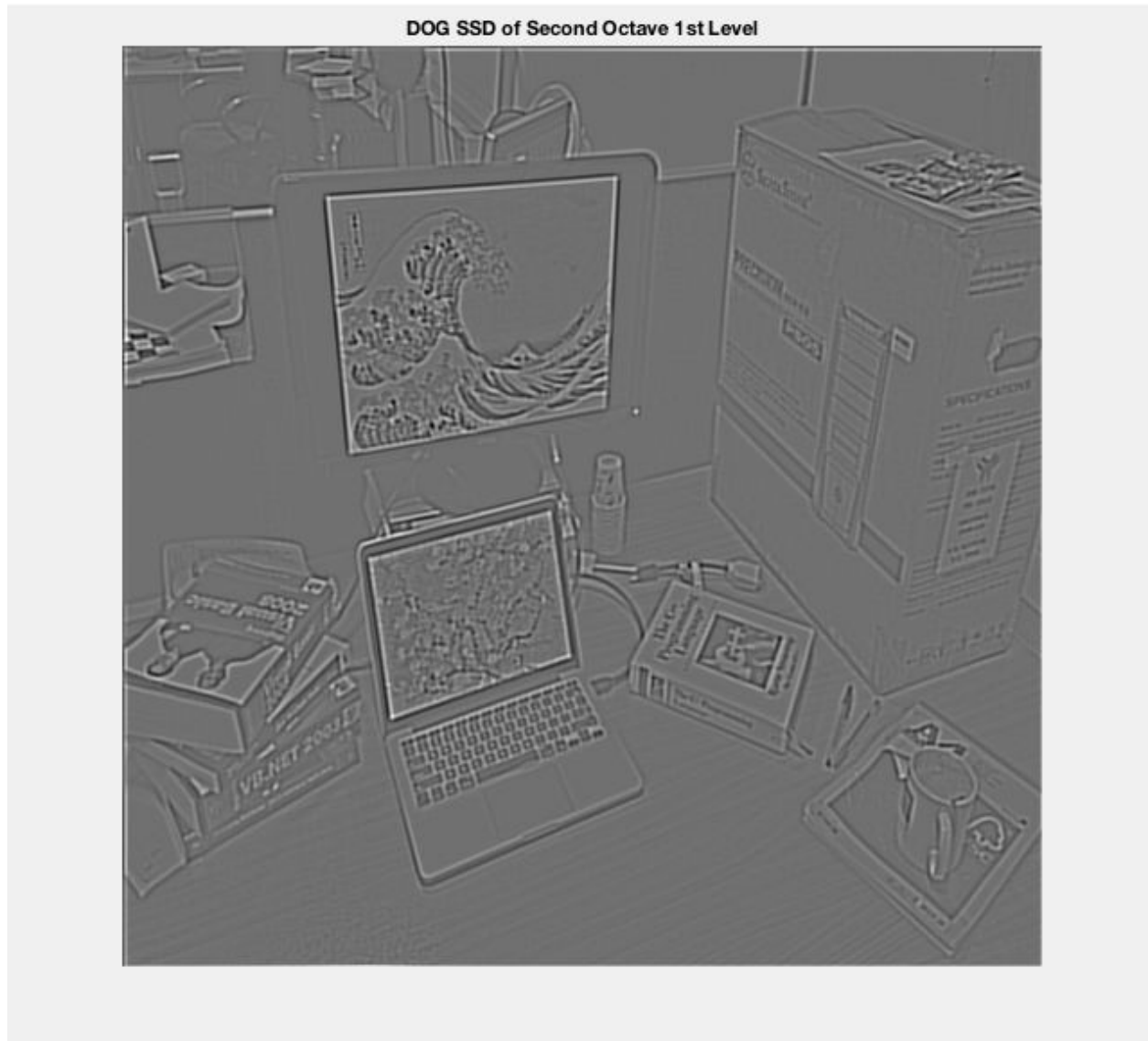


Fourth Level :

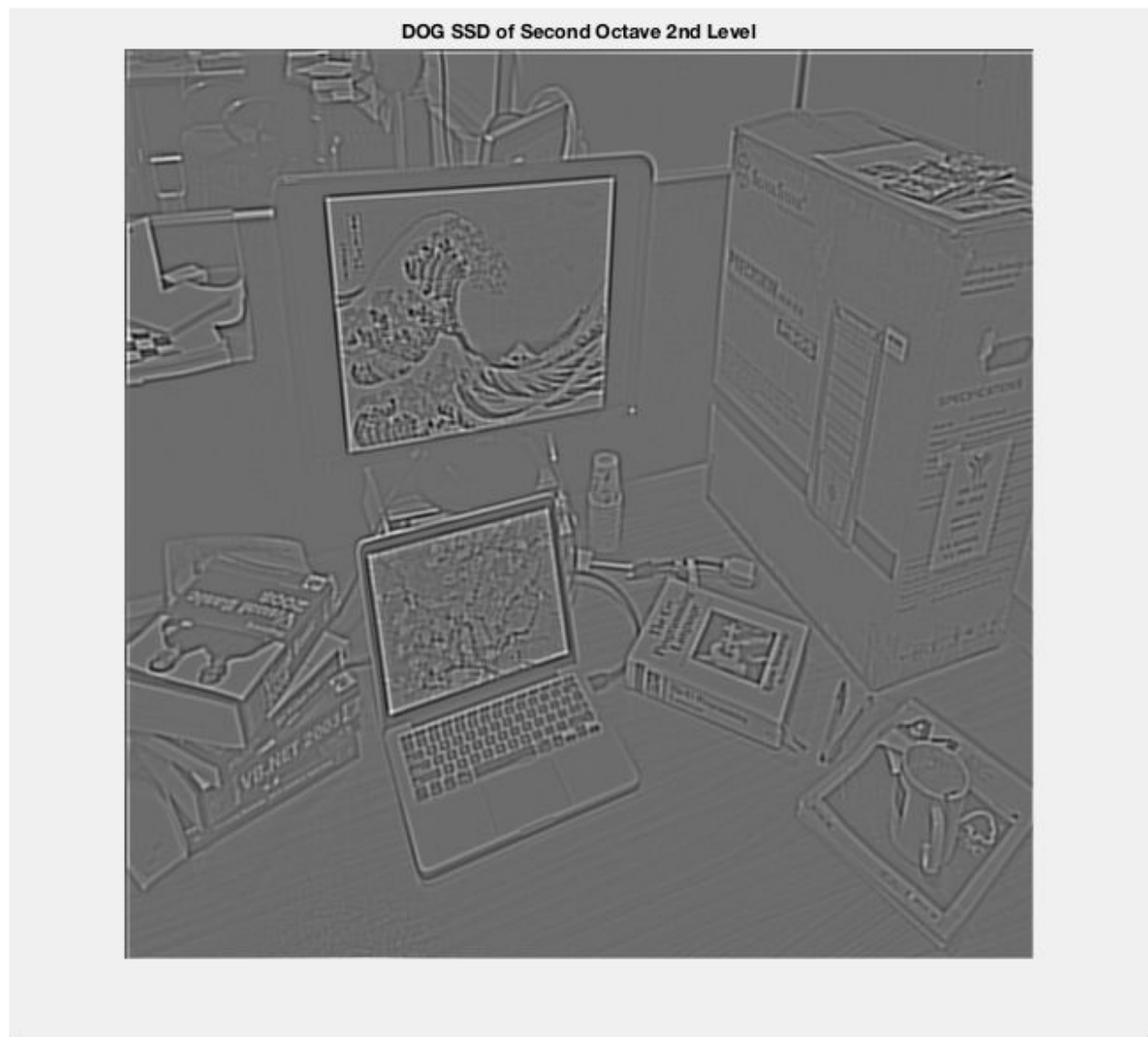


## 2. Octave 2

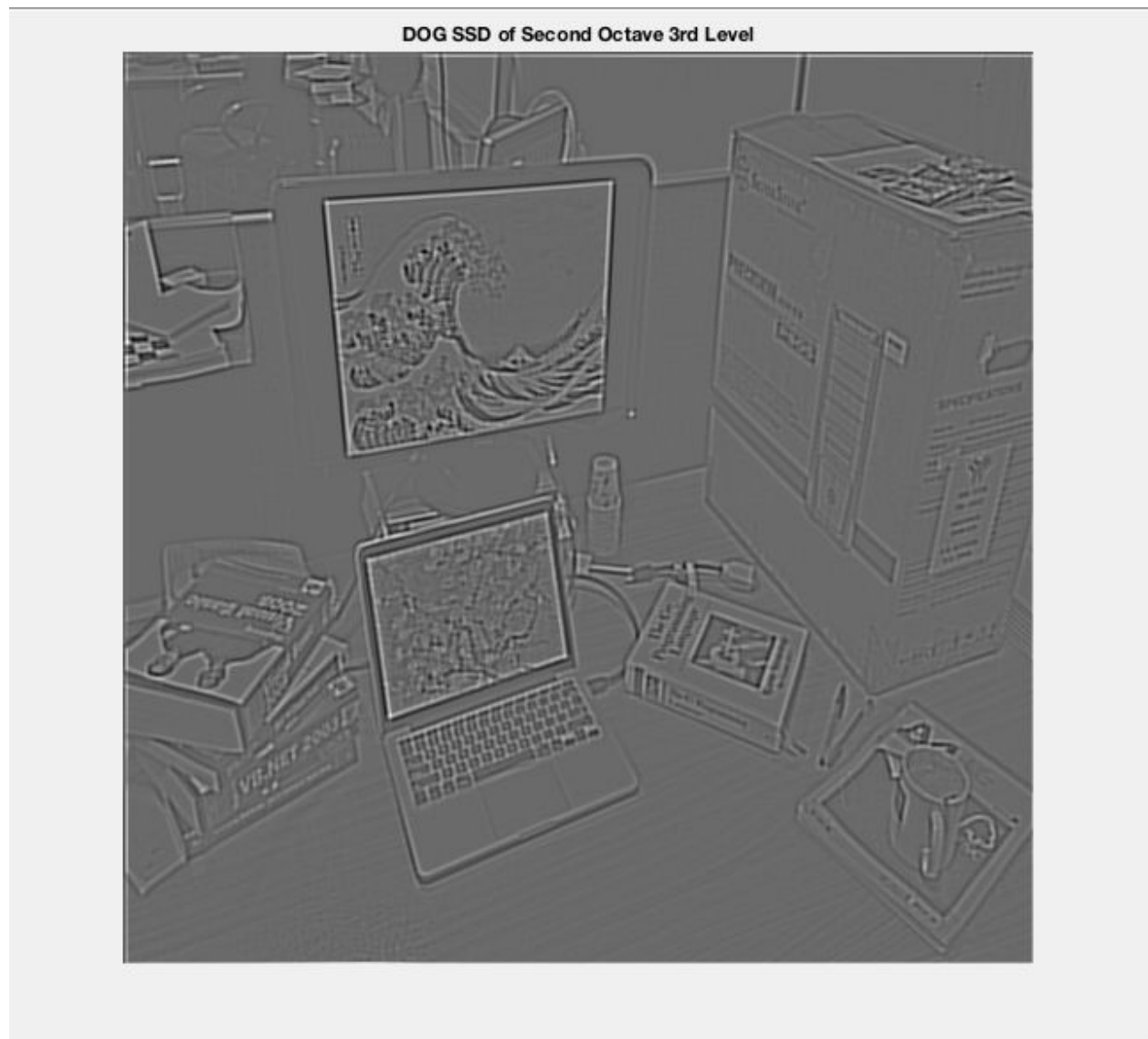
First Level :



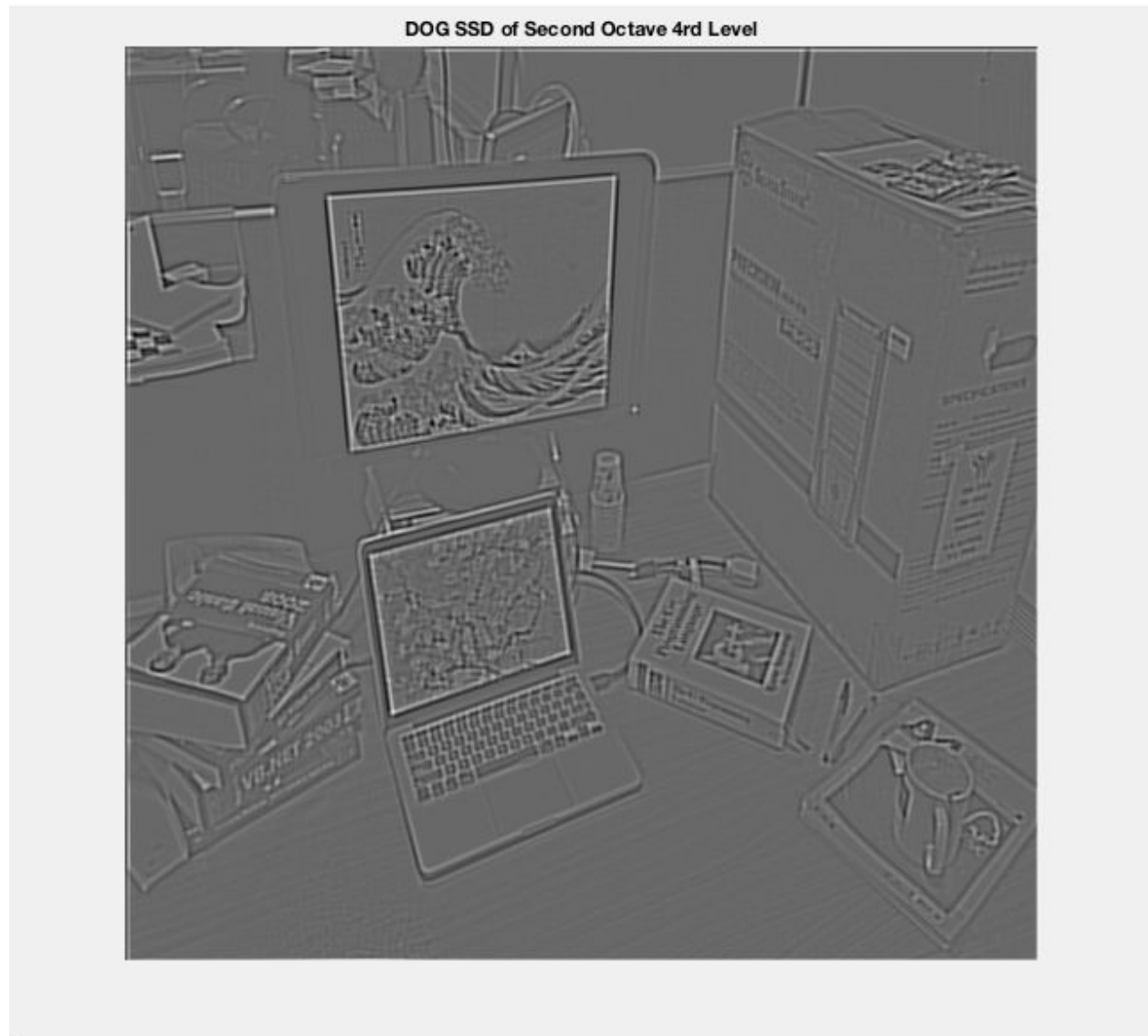
Second Level :



Third Level :



Fourth Level :



### 3. Octave 3

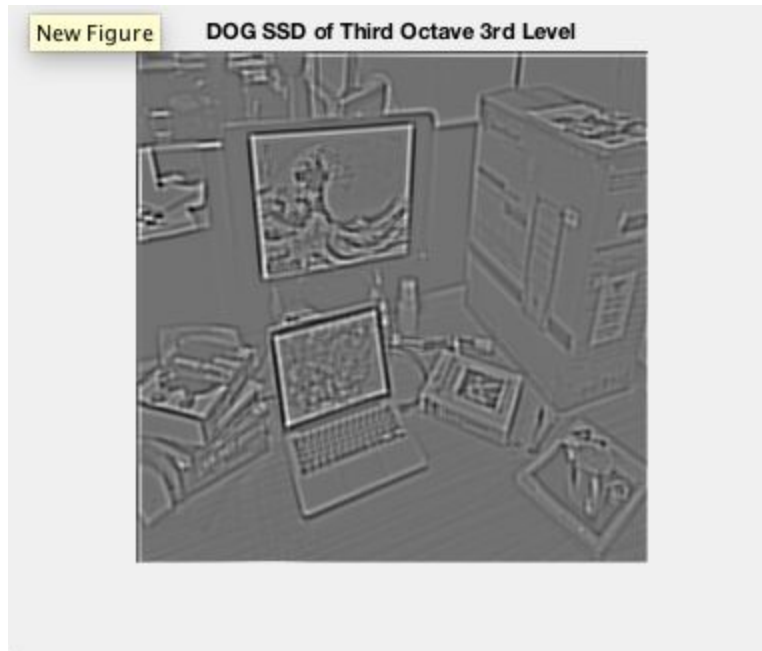
First Level :



Second Level :



Third Level :



Fourth Level :



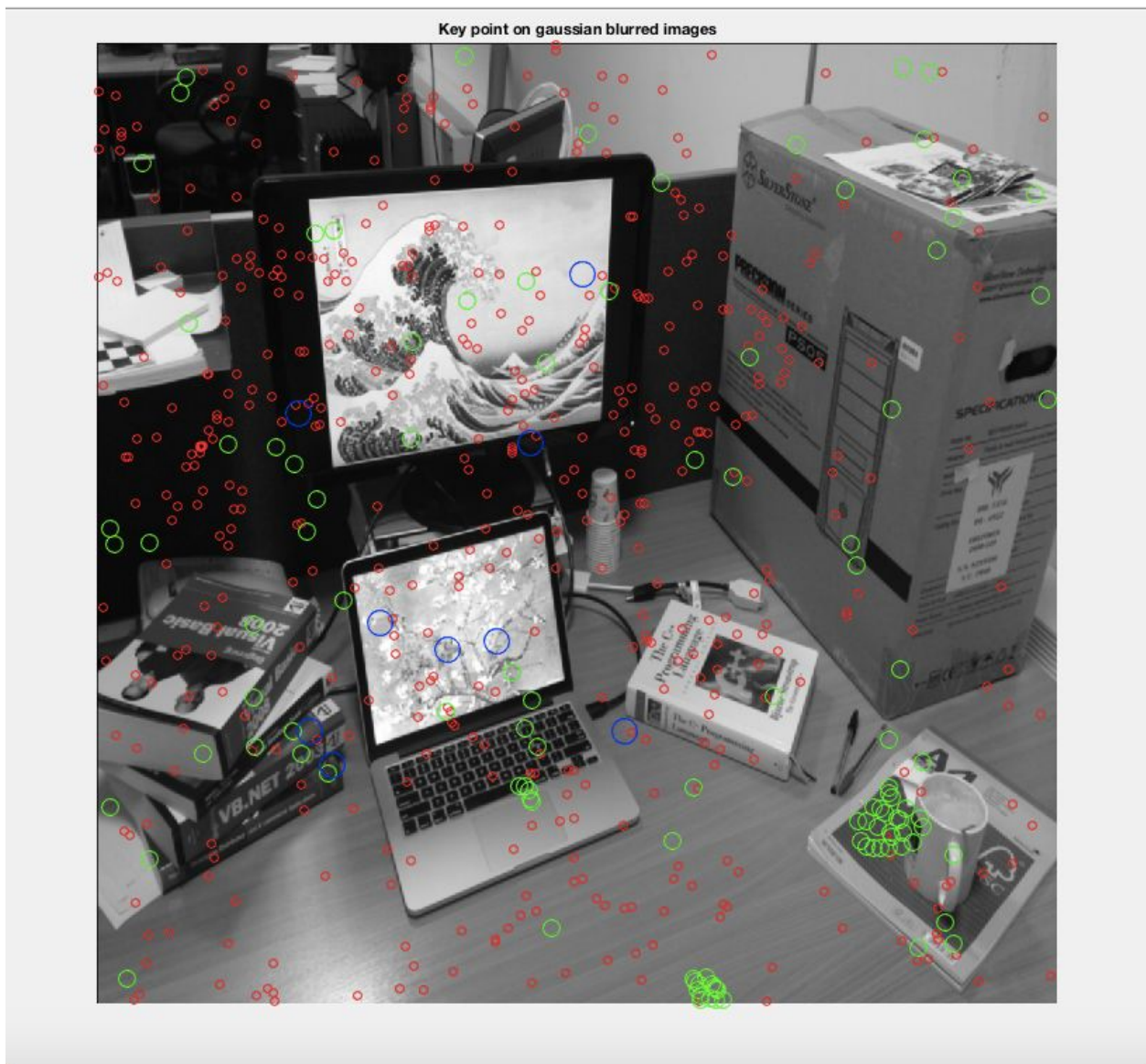


## b) Keypoint detection

Find the extrema from all three layer(Top, Bottom and Middle layer) from each octave.

a) Low contrast Rejection  $|D(x)| < th$

b) Spurious Edge Rejection  $r > 10$



Images highlighting detected keypoints on the Gaussian blurred images

c) Orientation assignment

- Smooth by Gaussian of 1.5 times actual sd.
- For each key point in L, consider a 16x16 neighbourhood
- Split into 36 bins of 10 deg each.
- Weight angle theta by magnitude mag.
- Take orientation greater than 80% of max magnitude.



Images highlighting the dominant orientation of the keypoints.

d) Keypoint descriptor

- Convolve with 2D gaussian of sigma =0.5 x 16 on  $I(x,y)$
- Construct 8 bin histogram on every 4x4 region.
- Concatenate 16 bin each of size 8.
- Descriptor size =  $16 \times 8 = 128$  dim.

Store data in feature1 vector of size 130x596 in which last two dimensions for data point, which I used while matching the point.

Feature\_I1 =feature1 of dimension 128x596

Feature\_I2 =feature2 of dimension 128x689

Original Image 2: Inverted Image 3\_2.jpg



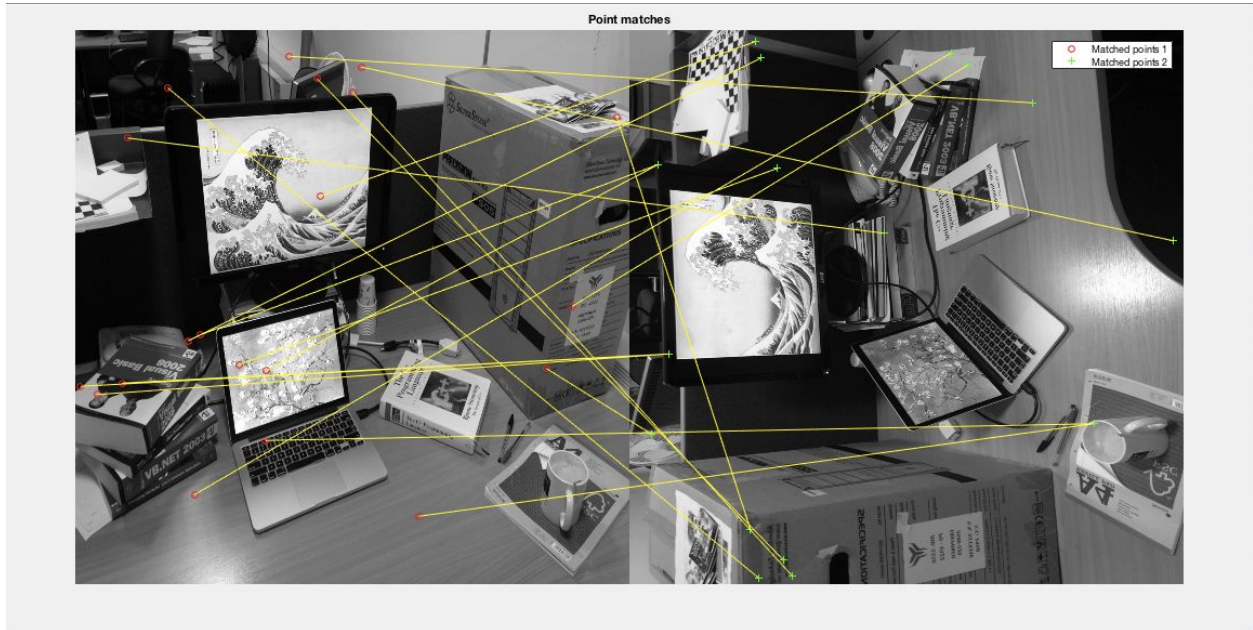
I also made all the steps like this old image for this image.(Refer Code)



(e). Image matching

Image matching(feature1,feature2)

Find the Euclidean distance between both feature descriptor and sort in ascending order and plot top 20 match feature in both the images.



20 keypoints matched between two images using minimum distance approach