HARRIS CORNER DETECTOR

DEEPAK PANT
2017CSB1072

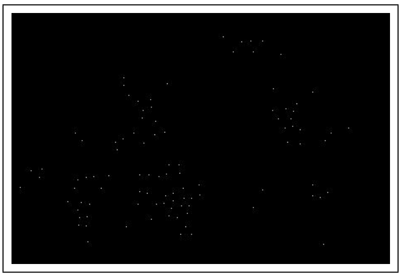
Indian Institue of Technology Ropar

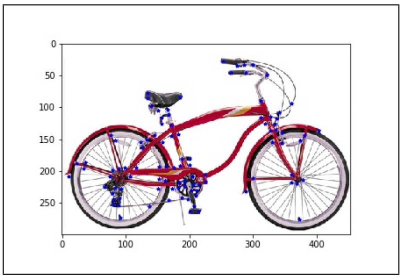
Abstract

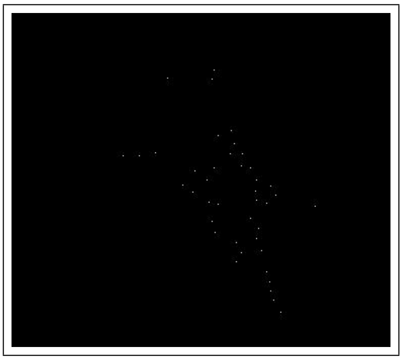
Harris corner detector is an algorithm used to detect corners or interest points in an image. In this assignment, I implemented a version of harris corner detector which uses the corner response function present in the original paper.

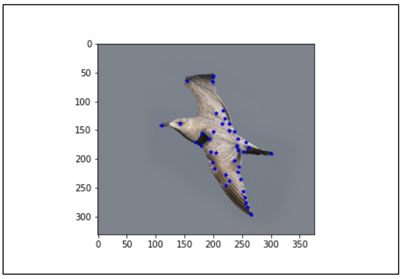
A corner is a point whose local neighborhood include two edges. Corners are the important features in the image, and are also known as interest points.
Firstly, compute the grayscale image. Then find smoothed gradients. Then for each pixel find its corner response in a window around that pixel. Then I found all pixels above a threshold value, and applied non-maximal suppression to remove blobs of same corner.

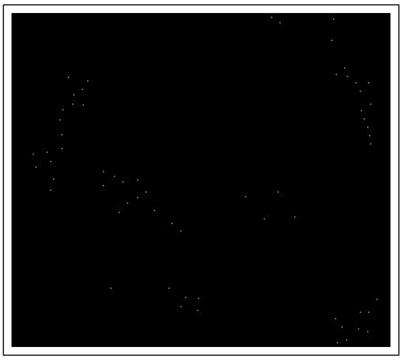
1 Output Images

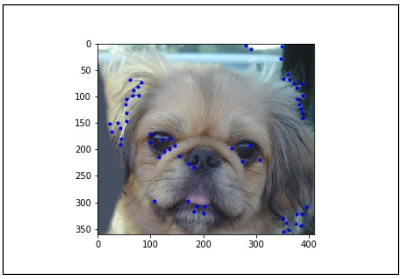


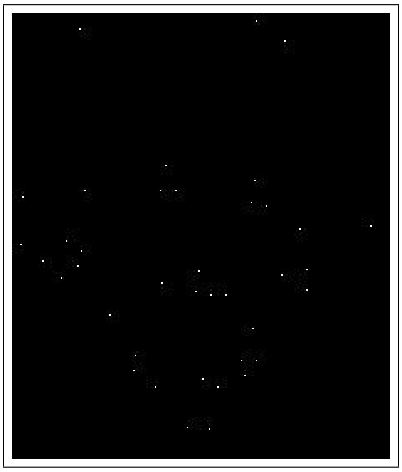


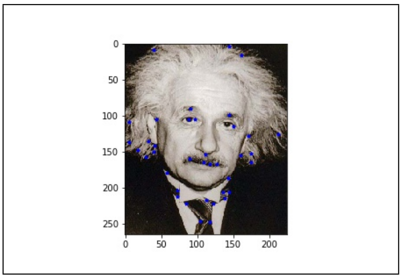




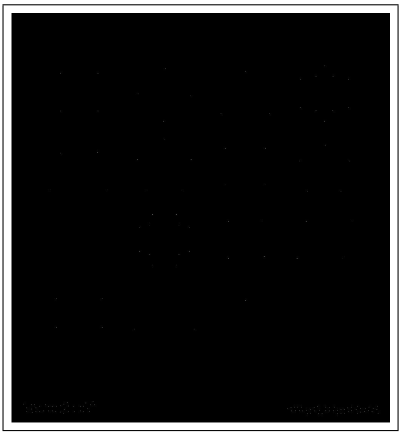


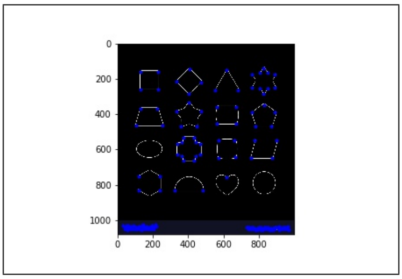






2 Algorithms





Algorithm 1 find-corners(input-img)

```
1:  Find luminance of the input image.
2:
3:  Find gradient in x and y direction using sobel filter.
4:  $fxgrad = gradient\,along\,x\,direction$
5:  $fygrad = gradient\,along\,y\,direction$
6:  arr contains x, y Co-ordinates and its corner response
7:
8:  **while** $i < height$ **do**
9:      **while** $j < width$ **do**
10:         In window of size 9 * 9 around i, j find determinant and trace of
11:         covariance matrix
12:         R = determinant - (0.04 * trace * trace)
13:         Add this i,j,R to arr
14:     **end while**
15: **end while**
16:
17: L contains x, y co-ordinate(whose value is greater than threshold) and their corner re-
    sponse of those co-ordinates.
18: threshold = ratio * max(R)
19:
20: **while** $res < arr.length$ **do**
21:     i, j, R = res
22:     **if** $R > threshold$ **then**
23:         L.append([i, j, R])
24:     **end if**
25: **end while**
26: return L
```

Algorithm 2 NonMaximalSuppressor(L)

```
1:  sortedL contains L sorted in decreasing order of R value
2:  finalL contains list after non maximal suppression
3:  Insert first element of sortedL in finalL
4:
5:  while $i\ in\ sortedL.length$ do
6:      while $j\ in\ finalL.length$ do
7:          If i is present in a window of side dis around j.
8:          break
9:
10:     end while
11:     Include this co-ordinate in finalL
12: end while
13:
14: Mark these co-ordinates as Corners
15: Overlay these coordinates with input image for representation.
```

3 Observations

Harris corner detector shows good output with correct thresholding and implementing some heuristic to remove blobs of corners. The threshold varied quite strongly in case of different images.

Firstly, I calculated corner response of the whole image. Then I extracted pixels which were over a predefined threshold. For non maximal suppression, I also removed pixels which were not just in the 8-connected neighbourhood but a box of 8-connected pixels. This prevent "pierced blobs" in the resulting corner image.

In the end I overlayed the images using matplotlib.pyplot.plot function. The image which are slightly curved easily gets eliminated by harris corner detector.

4 Conclusion

In this assignment, I implemented Harris corner detector with corner response function present in the original paper(cr = det - k*trace**2). In it I also used a non-maximal suppression to remove points around these corners. Further I used a box to remove blobs of corner in the image. At the end I overlayed both images for better representation.