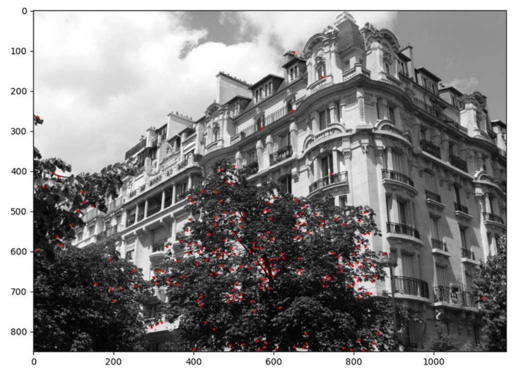
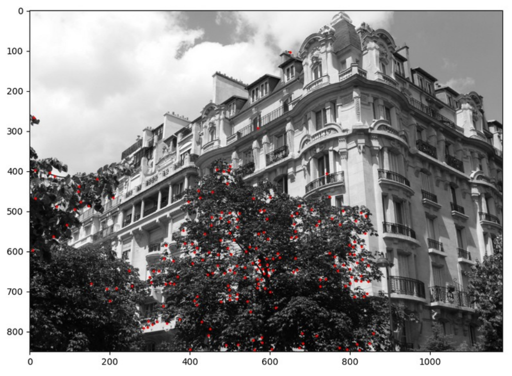
HOMEWORK SET 6 – Víctor Mira Ramírez

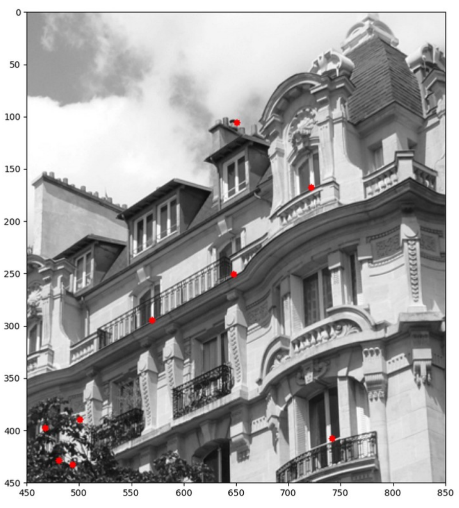
Exercise 1

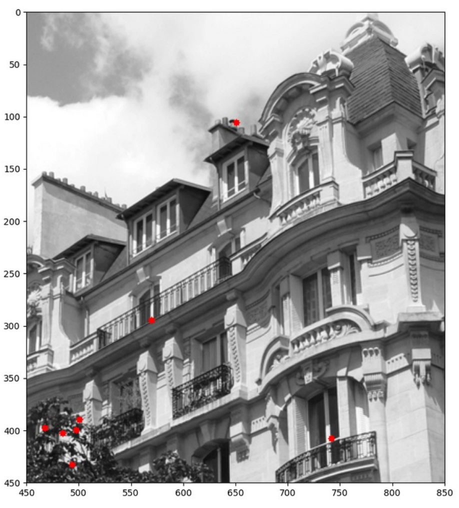
To solve this exercise, both Shi-Tomasi and Harris corner detector methods were used. A copy of the grayscale image was used to obtain both method’s corners. Then they were printed on the original image with red dots. We can see the obtained image for the Shi Tomasi corner detector ond the left and Harris on the right.





We can zoom out into the image to further investigate on the differences between these methods:





As we can see the Shi-Tomasi picked more consistenly the points on the ledge of the buliding, while the Harris picked more on the tree leaves. I consider the points on the leaves to be less useful as they seem to be less distinct for this image. In this case, with the limitations of the exercise, I think Shi-Tomasi corner detector does a better job.

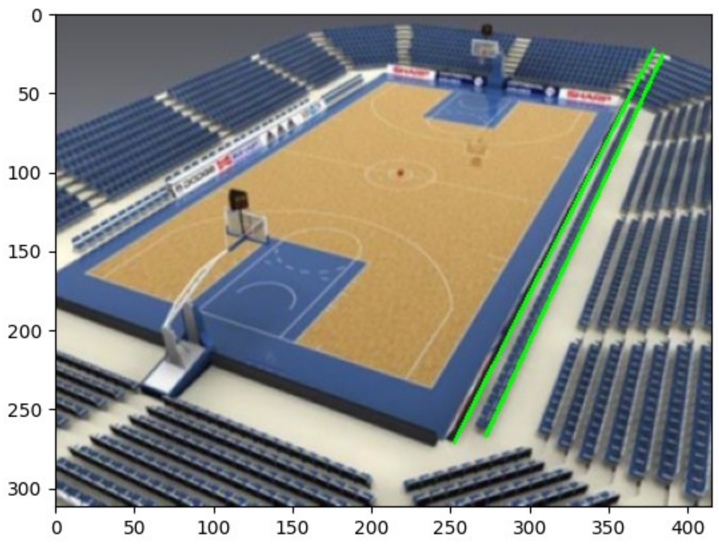
Code:

```
path=r'/home/victor/fisicaua/tercero/SIUE/robotic_vision/entregas/hw6/building2-1.png'


img = cv2.imread(path, cv2.IMREAD_COLOR)
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# SHI-TOMASI
shi_tomasi = img.copy()
corners = np.int0(cv2.goodFeaturesToTrack(gray.copy(), 200, 0.01, 10))
for i in corners:
x, y = i.ravel()
cv2.circle(shi_tomasi, (x, y), 3, (255, 0, 0), -1)

# HARRIS
harris = img.copy()
corners = np.int0(cv2.goodFeaturesToTrack(gray.copy(), 200, 0.01, 10,
useHarrisDetector=True,k=0.1))
for i in corners:
x, y = i.ravel()
cv2.circle(harris, (x, y), 3, (255, 0, 0), -1)

plt.figure()
plt.imshow(shi_tomasi)

plt.figure()
plt.imshow(harris)
```

Exercise 2

For this exercise I was unable to capture images with my laptop because it doesn't have one. I am not sure of what was I supposed to do without one, so I did the rest of the exercise with an image from a previous homework.

As we can see, houghlines method gives us a list of detected lines given a few parameters that can be tinkered with to obtain the desired result. We then print the first two, most predominant lines on the original image and display it. We can change the colors of the line on the method cv2.line().



Code:

```
path=r'/home/victor/fisicaua/tercero/SIUE/robotic_vision/entregas/hw4/
basketball_court.jpg'
img = cv2.imread(path, cv2.IMREAD_COLOR)
rgb_img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Use canny edge detection
edges = cv2.Canny(gray,50,150,apertureSize=3)
lines_list =[]
lines = cv2.HoughLinesP(edges, 1, np.pi/180,  threshold=200,  minLineLength=5,
maxLineGap=10)

x1,y1,x2,y2=lines[0][0]
cv2.line(rgb_img,(x1,y1),(x2,y2),(0,255,0),2)
x1,y1,x2,y2=lines[1][0]
cv2.line(rgb_img,(x1,y1),(x2,y2),(0,255,0),2)
plt.figure()
plt.imshow(rgb_img)
```