

Problems

- Estimate image gradients and apply the Harris corner detection algorithm
- Obtain a better localization of each corner
- Compute a feature vector for each corner point
- Display the corners by drawing empty rectangles over the original image
- Use the feature vectors to match the feature points

Solution

- Load two similar images
- Convert to grayscale and smooth the image
- Compute gradient vectors
- For each point, calculate Harris score and threshold the score
- If a point is a corner, apply corner localization to find location of the corner
- Apply SIFT feature detector to images to find key points.
- Search in the list of key points for the corner location found previously. Thus, we get the feature vector for the corner points
- Apply brute force matcher to find matches of corner points in two similar images
- Interactively controlled parameters are implemented as input through keyboard

Implementation

Function cornerH (image, s, winSize, k, threshold)

- Use cv2.sobel to get gradient vectors
- Calculate Harris score by $HC = \det(C) + k * (\text{trace}(C)^2)$ for each window
- If a corner point is found, apply the formula $p^* = \text{inv}(C) * \sum g_i * g_i.T * x_i$
- Put corner points and corner location in two lists

Function drawRectangle(image, corners, kp)

- For each corner point, draw a blue rectangle around it and red dot in the center
- For each corner location, draw a green dot

featureVector(image1, image2, kp1, kp2)

- Using SIFT feature detector, we have a list of key points. Search in the list for corner points found in corner detection.
- Use brute force matcher to find similar corner points in two images
- Number corresponding points

Configuration

- To run the program, open a terminal, go to the directory “.../src” and type “python AS2.py”
- Make sure to run the corner detection first by typing “c” in the program

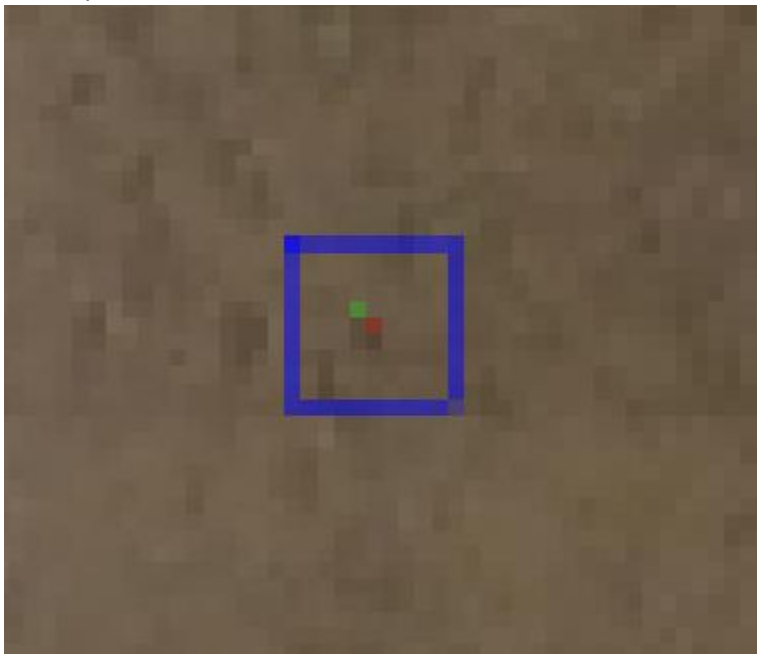
```
h
'c': Find corners and better location
'f': Compute feature vector for corner and find matches
'q': Exit
c
the variance of Gussian:3
windowSize :3
the weight of the harris conner detector:0.04
threshold:1000000000
Computing...
Draw...
```

-
- Then run the feature matching by typing "f"

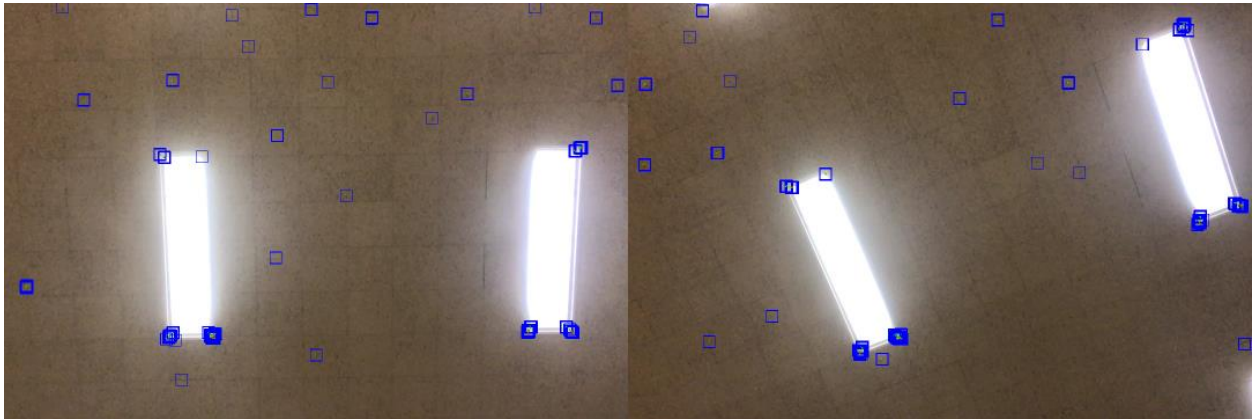
Results

The red points are the center of windows where corner is detected.

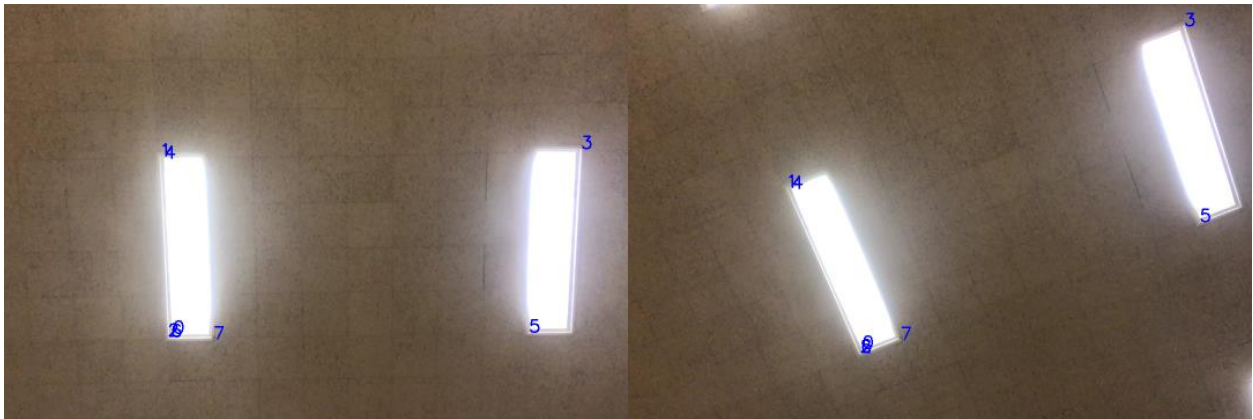
Green points are better localization of corners.



Blue rectangles are the pixel that corner detected.



Matching corners are given identical numbers.



Helper function

```
h
'c': Find corners and better location
'f': Compute feature vector for corner and find matches
'q': Exit
```

Interactive parameters

```
h
'c': Find corners and better location
'f': Compute feature vector for corner and find matches
'q': Exit
c
the variance of Gussian:3
windowSize :3
the weight of the harris conner detector:0.04
threshold:1000000000
Computing...
Draw...
```

