

CS512 Assignment 2:Report
Prateek Parab
Department of Computer Science
Illinois Institute of Technology
October 22,2017

Abstract

This programming assignment deals with corner detection using the Harris corner detection algorithm. Here two images are loaded separately and corner detection algorithm is applied to both the images. Thus feature matching will be done on both the images although the images won't be exactly same but similar. The corners detected will be displayed in a final image along with the the matched features presented in two separate images. The parameters can be changed using a trackbar functionality.. Corner detection is performed in openCV.

Problem Statement:

The program should be designed to perform corner detection for a given image. It should load an image by reading it from a test image provided by the user. The user should be allowed to perform manipulation to the image by pressing specific keys on the keyboard such as a help key. It also allows the user to control the feature matching parameters using a trackbar functionality.

Proposed Solution:

The program is developed to perform corner detection for a specified image. To perform this functions user must press specific keys. The program comprises of loops and a separate algorithm for corner detection other than the inbuilt openCV function. The images have been tested according to their various resolutions and corners are detected and displayed using windows and then the recorded corners are numbered accordingly when they are compared with a similar image which is done by comparing their feature vectors.

Implementation Details:

In this program corner detection is performed using Harris corner detection. First the window size, corner response and threshold is taken as an input from the user. An offset is calculated based on the window size. Next step is to calculate the derivate in the x direction , y direction and the xy direction. The obtained values are summed up using a summation function in OpenCV. A determinant is calculated along with a trace parameter by add the derivative in x and y direction respectively. Later a corner response value is generated based on the trace and determinant value and also the user input corner response value. If the generated corner response value is greater than the provided threshold then note the corner point and color the point. Thus Harris corner detection is performed

Results:

A final image is obtained based on the corner detection operation performed along with matched features. These features are numbered according to their similarities in the respective images. The features are varied by changing few parameters using the trackbar functionality.

Screenshots:

1) Corners Detected for a test image with variable parameters(Image of hexagons)

The image displays two screenshots of the Spyder Python IDE, showing the results of corner detection on a test image (a hexagonal pattern) using the Harris corner detection algorithm. The code in the editor defines a function `compute_harris` that takes a frame, sigma, and k as inputs. The results are visualized in two windows: 'corner' and 'cornerDetectionHarris.py'.

Top Screenshot: The 'corner' window shows the results for a threshold of 10. The 'cornerDetectionHarris.py' window shows the results for a threshold of 10. The 'corner' window displays the detected corners as red dots on the image. The 'cornerDetectionHarris.py' window displays the detected corners as red dots on the image.

Bottom Screenshot: The 'corner' window shows the results for a threshold of 9. The 'cornerDetectionHarris.py' window shows the results for a threshold of 9. The 'corner' window displays the detected corners as red dots on the image. The 'cornerDetectionHarris.py' window displays the detected corners as red dots on the image.

The code in the editor is as follows:

```
1# -*- coding: utf-8 -*-
2"""
3Created on Fri Oct 20 01:12:32 2017
4
5@author: Prateek
6"""
7
8
9import cv2
10import sys
11import numpy as np
12import matplotlib.pyplot as plt
13
14
15
16
17
18
19
20def nothing():
21    pass
22
23
24
25def compute_harris(frame, sigma, k):
26
27
28    rows = frame.shape[0]
29    cols = frame.shape[1]
30
31    cov = np.zeros((rows,cols * 3), dtype = np.float32)
32    dst = np.zeros((rows,cols), dtype = np.float32)
33    #computing x and y derivative of image
34    dx = cv2.Sobel(frame,cv2.CV_64F,1,0,ksize=5)
35    dy = cv2.Sobel(frame,cv2.CV_64F,0,1,ksize=5)
```

D:\homework and assignments\data intensive computing\Project\Pandas - Spyder (Python 3.6)

File Edit Search Source Run Debug Consoles Projects Tools View Help

Project explorer

- Pandas
 - pima-indians-d
 - test.py

Editor - D:\homework and assignments\computer vision\opencv python\cornerDetectionHarris.py

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Fri Oct 20 01:12:32 2017
4
5 @author: Prateek
6 """
7
8 import cv2
9 import sys
10 import numpy as np
11 import matplotlib.pyplot as plt
12
13
14
15
16
17
18
19
20 def nothing():
21     pass
22
23
24
25 def compute_harris(frame, sigma, k):
26
27     rows = frame.shape[0]
28     cols = frame.shape[1]
29
30     cov = np.zeros((rows,cols * 3), dtype = np.float32)
31     dst = np.zeros((rows,cols), dtype = np.float32)
32     #computing x and y derivative of image
33     dx = cv2.Sobel(frame,cv2.CV_64F,1,0,ksize=5)
34     dy = cv2.Sobel(frame,cv2.CV_64F,0,1,ksize=5)
```

IPython console

```
threds total 235668605239.0
threshold value 9
dst.max 2.61854e+13
threds total 235668605239.0
threshold value 9
dst.max 2.61854e+13
threds total 235668605239.0
```

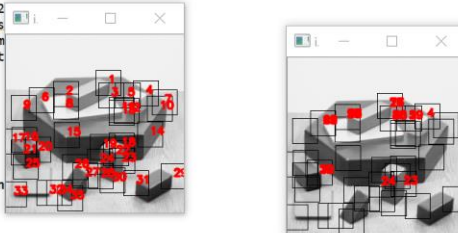
corner

Variance: 0

Neighb...ze: 1

Trace: 13

Threshold: 9



D:\homework and assignments\data intensive computing\Project\Pandas - Spyder (Python 3.6)

File Edit Search Source Run Debug Consoles Projects Tools View Help

Project explorer

- Pandas
 - pima-indians-d
 - test.py

Editor - D:\homework and assignments\computer vision\opencv python\cornerDetectionHarris.py

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Fri Oct 20 01:12:32 2017
4
5 @author: Prateek
6 """
7
8 import cv2
9 import sys
10 import numpy as np
11 import matplotlib.pyplot as plt
12
13
14
15
16
17
18
19
20 def nothing():
21     pass
22
23
24
25 def compute_harris(frame, sigma, k):
26
27     rows = frame.shape[0]
28     cols = frame.shape[1]
29
30     cov = np.zeros((rows,cols * 3), dtype = np.float32)
31     dst = np.zeros((rows,cols), dtype = np.float32)
32     #computing x and y derivative of image
33     dx = cv2.Sobel(frame,cv2.CV_64F,1,0,ksize=5)
34     dy = cv2.Sobel(frame,cv2.CV_64F,0,1,ksize=5)
```

IPython console

```
threds total 4.0351309014e+12
threshold value 54
dst.max 7.47246e+13
threds total 4.0351309014e+12
threshold value 54
dst.max 7.47246e+13
threds total 4.0351309014e+12
```

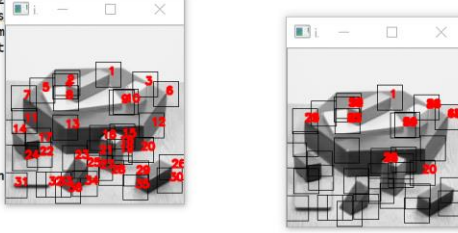
corner

Variance: 0

Neighb...ze: 1

Trace: 3

Threshold: 54



D:\homework and assignments\data intensive computing\Project\Pandas - Spyder (Python 3.6)

File Edit Search Source Run Debug Consoles Projects Tools View Help

D:\homework and assignments\computer vision\opencv python

Project explorer

- Pandas
 - pima-indians-d
 - test.py

Editor - D:\homework and assignments\computer vision\opencv python\cornerDetectionHarris.py

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Fri Oct 20 01:12:32 2017
4
5 @author: Prateek
6 """
7
8
9 import cv2
10 import sys
11 import num
12 import mat
13
14
15
16
17
18
19
20 def nothing
21     pass
22
23
24
25 def compute_harris(frame, sigma, k) :
26
27
28     rows = frame.shape[0]
29     cols = frame.shape[1]
30
31     cov = np.zeros((rows,cols * 3), dtype = np.float32)
32     dst = np.zeros((rows,cols), dtype = np.float32)
33     #computing x and y derivative of image
34     dx = cv2.Sobel(frame,cv2.CV_64F,1,0,ksize=5)
35     dy = cv2.Sobel(frame,cv2.CV_64F,0,1,ksize=5)
```

Console 1/A

```
threds total 896695755866.0
threshold value 12
dst.max 7.47246e+13
threds total 896695755866.0
threshold value 12
dst.max 7.47246e+13
threds total 896695755866.0
```

corner

Variance: 0

Neighb...ze: 1

Trace: 3

Threshold: 12

dst.max 7.47246e+13
threds total 896695755866.0
threshold value 12

Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 1 Column: 1 Memory: 62 %

Type here to search

5:23 PM 10/22/2017

References:

1) Stack overflow

2) 2)opencv.org

3) <http://www.meccanismocomplesso.org/en/opencv-python-harris-corner-detection-un-metodo-per-rilevare-i-vertici-in-unimmagine>