Andrew Garcia
7/26/2022

ECE 5554 SU22 – Dr. Jones – HW3

Part 1d and 1g: Console Output

```
Top 3 corners using cv2.goodFeaturesToTrack

Filename: AllmanBrothers.png

(x,y): (246.0,39.0)

(x,y): (530.0,230.0)

(x,y): (628.0,68.0)

Top 3 corners using Harris Corner Method

Filename: AllmanBrothers.png

(x,y): (529,229)

(x,y): (495,77)

(x,y): (246,39)

Top 3 corners using cv2.goodFeaturesToTrack

Filename: CalvinAndHobbes.png

(x,y): (470.0,519.0)

(x,y): (422.0,556.0)

(x,y): (293.0,541.0)

Top 3 corners using Harris Corner Method

Filename: CalvinAndHobbes.png

(x,y): (538,380)

(x,y): (199,499)

(x,y): (532,454)

Top 3 corners using cv2.goodFeaturesToTrack

Filename: Chartres.png

(x,y): (307.0,748.0)

(x,y): (319.0,744.0)

(x,y): (318.0,680.0)
```

Andrew Garcia
7/26/2022

Top 3 corners using Harris Corner Method

Filename: Chartres.png

(x,y): (299,745)

(x,y): (315,750)

(x,y): (326,747)

Top 3 corners using cv2.goodFeaturesToTrack

Filename: Elvis1956.png

(x,y): (579.0,411.0)

(x,y): (616.0,426.0)

(x,y): (577.0,399.0)

Top 3 corners using Harris Corner Method

Filename: Elvis1956.png

(x,y): (617,426)

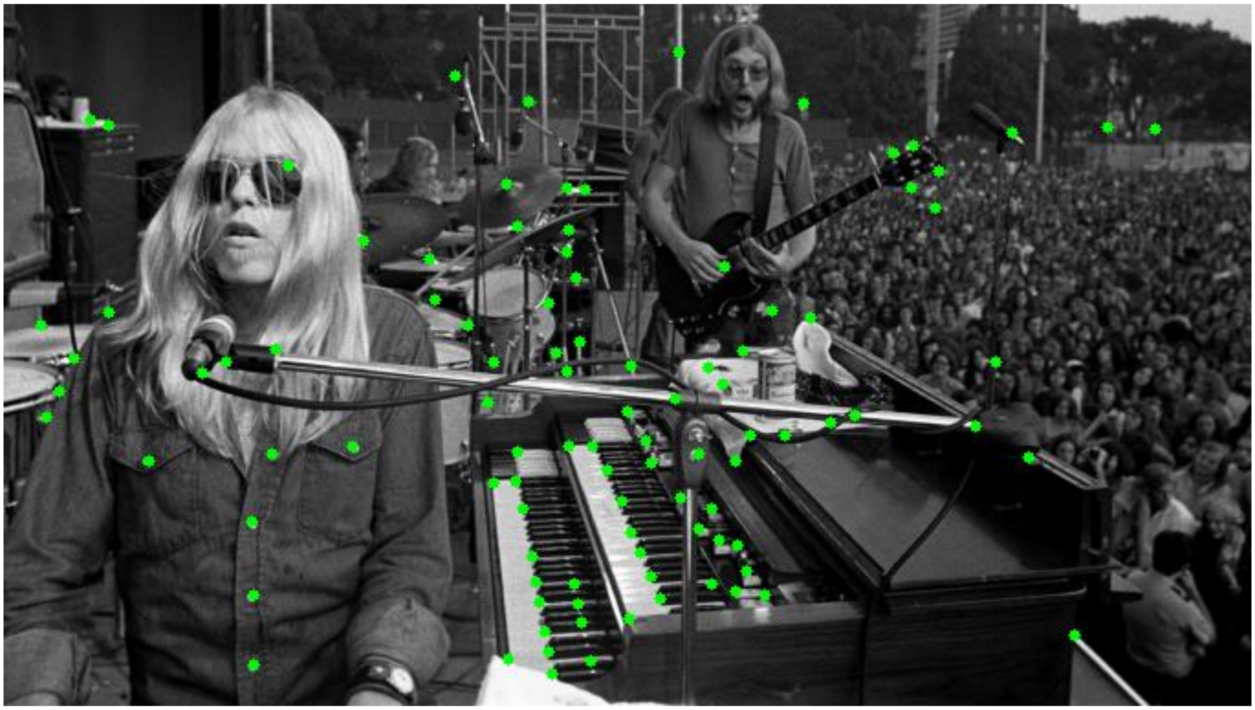
(x,y): (580,411)

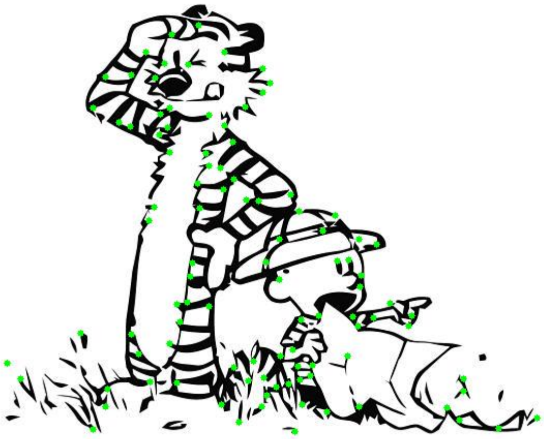
(x,y): (592,425)

Andrew Garcia
7/26/2022

Part 1e and 1h: Draw circles using both methods

cv2.goodFeaturesToTrack Method:





Andrew Garcia
7/26/2022

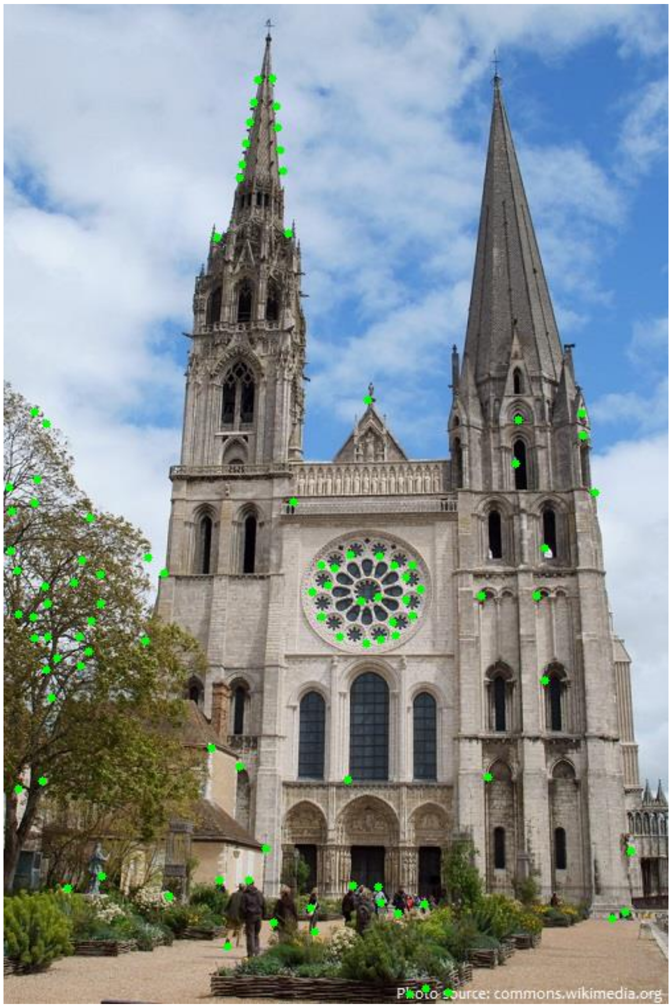
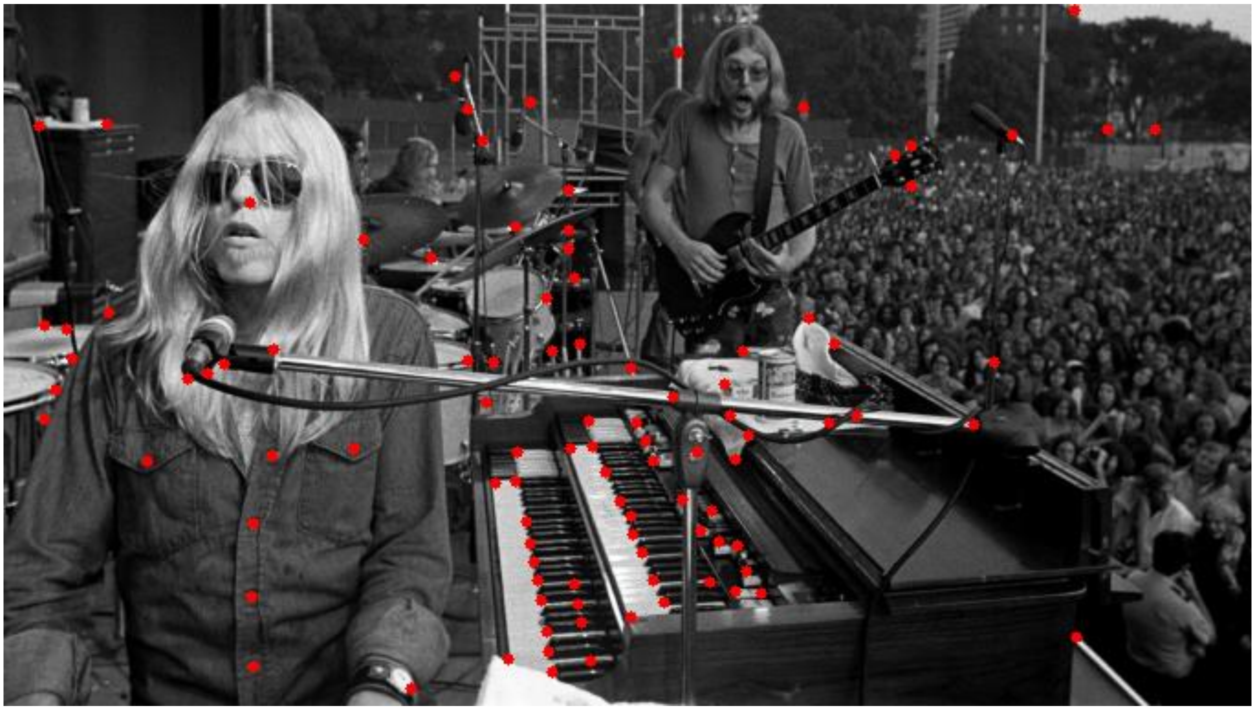


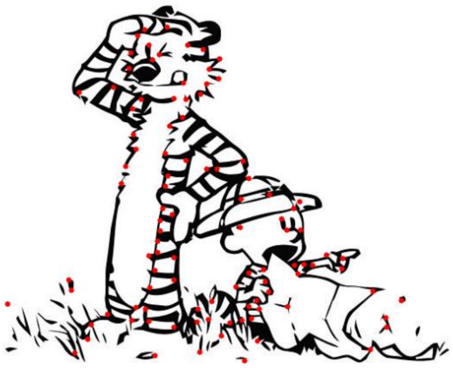
Photo source: commons.wikimedia.org

Andrew Garcia
7/26/2022

Andrew Garcia
7/26/2022

Implementation of Harris Corner Detector:





Andrew Garcia
7/26/2022

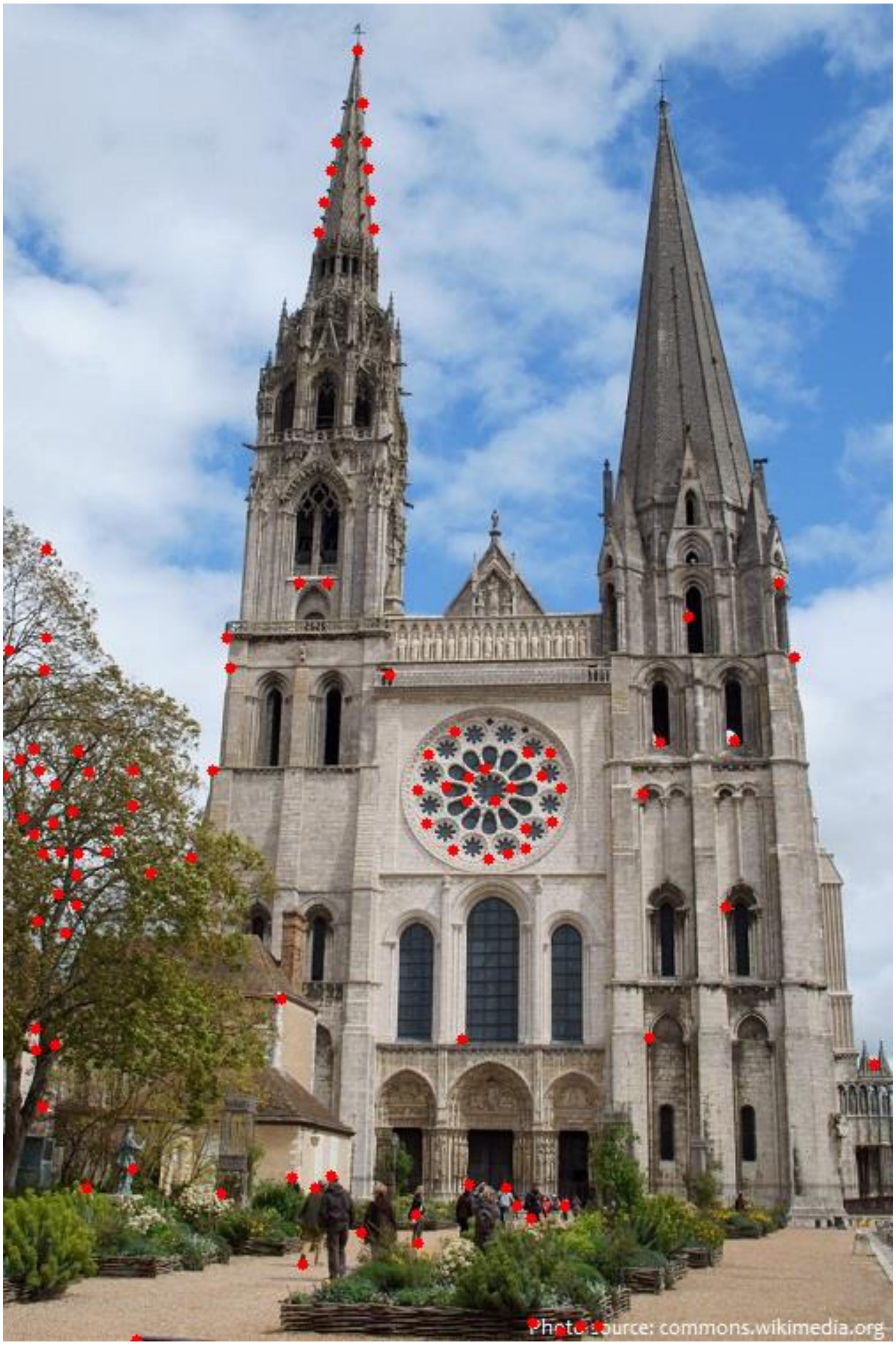
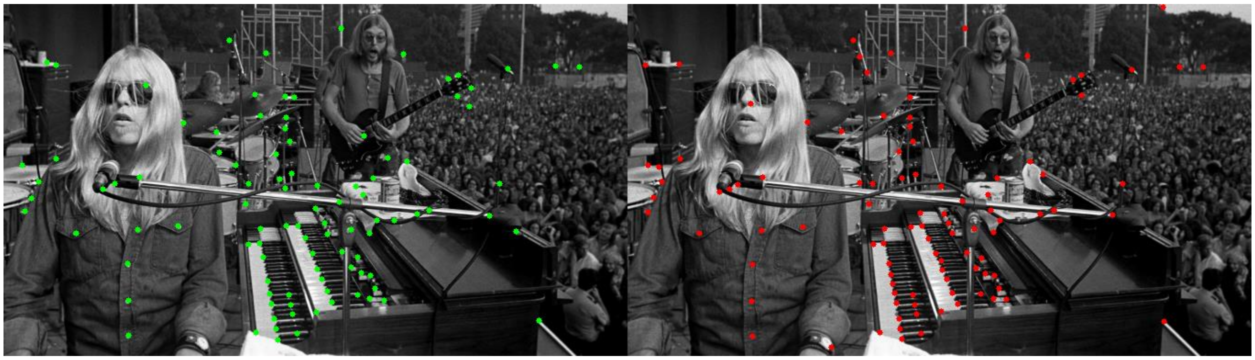


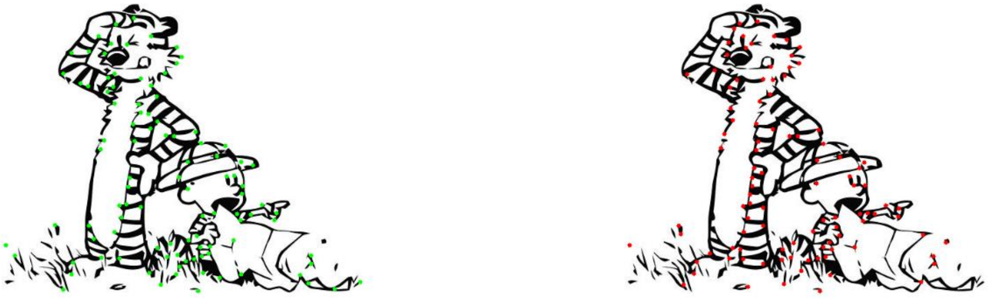
Photo source: commons.wikimedia.org

Andrew Garcia
7/26/2022

Andrew Garcia
7/26/2022

Side by side:





Andrew Garcia
7/26/2022

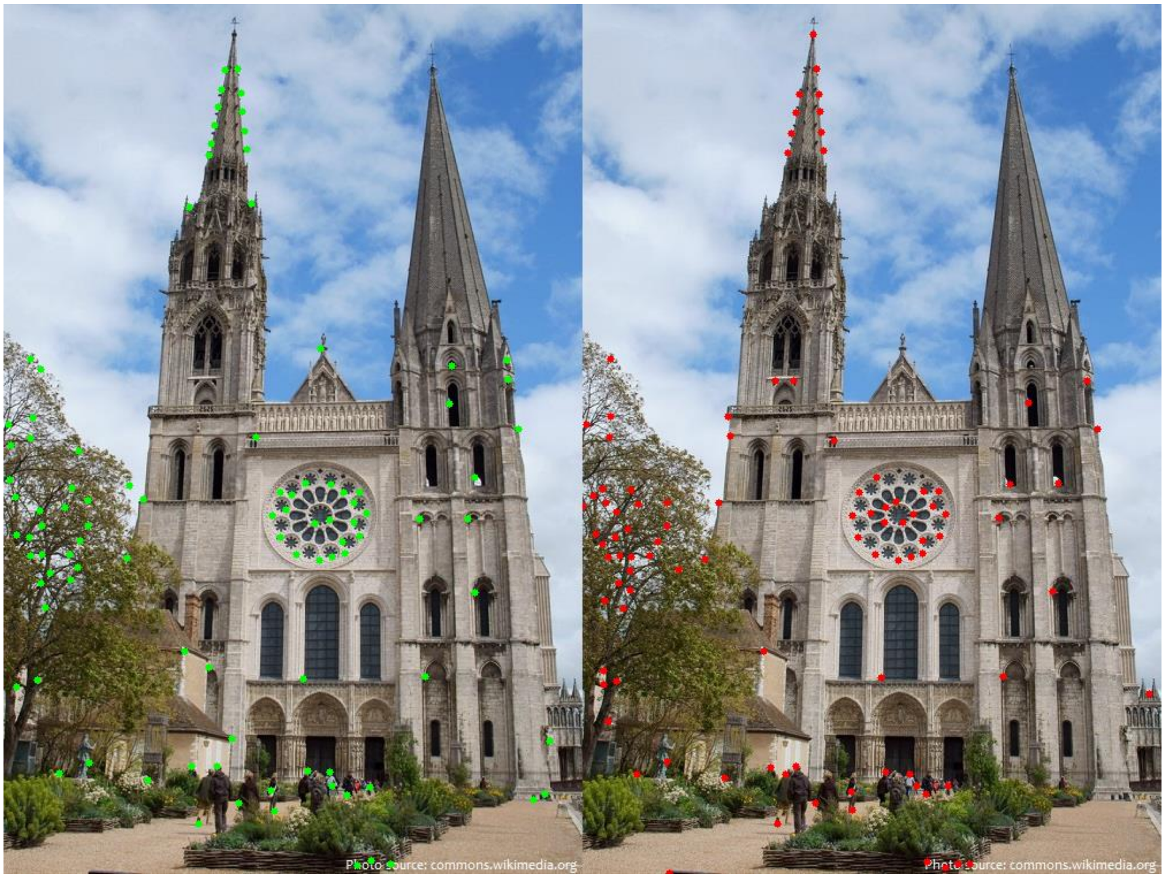
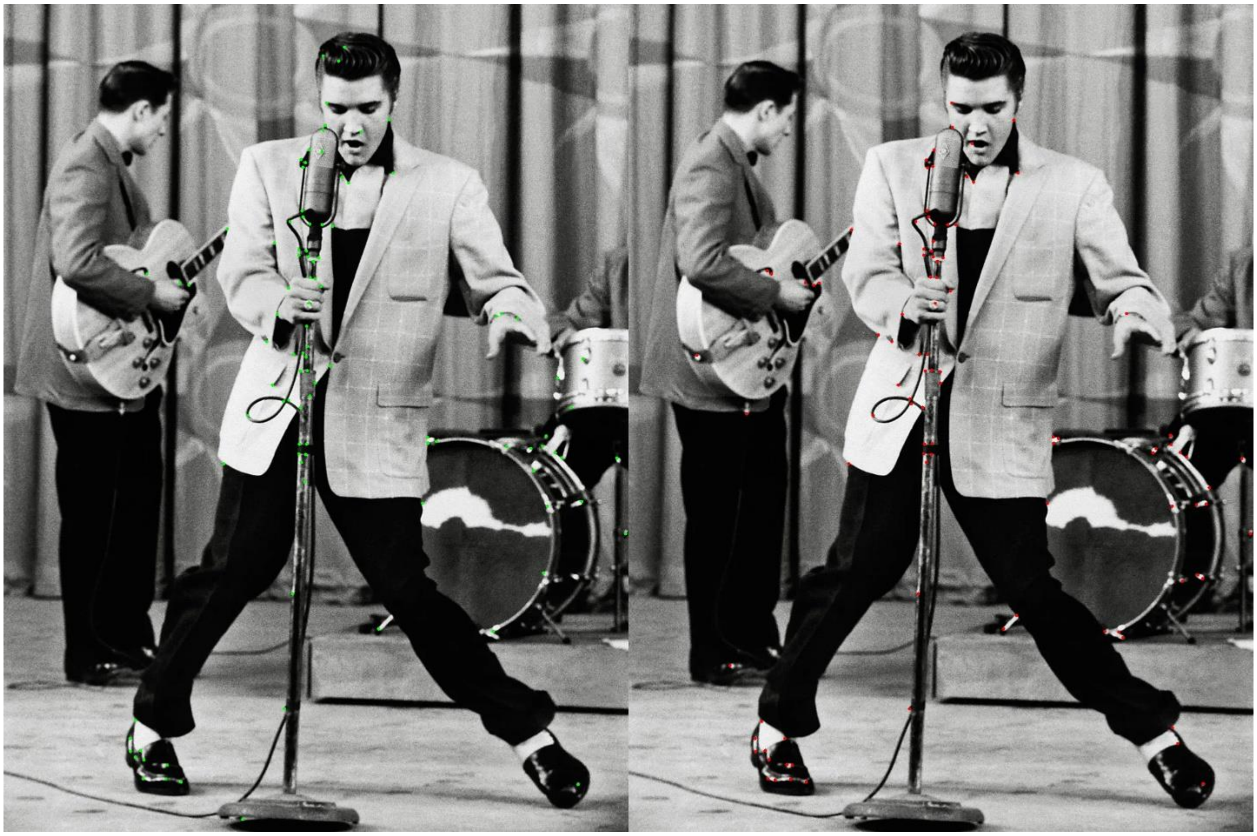


Photo source: commons.wikimedia.org
Photo source: commons.wikimedia.org



Andrew Garcia
7/26/2022

Discussion:

The parameters set for corner detection in both methods were the same, those parameters being max pixels of 100 with a pixel distance of 10 apart. For all 5 images, the top three intensities for corners where nearly identical or just ranked in a different order. For example, in AllmanBrothers.png the top-rated location for the cv2 function was the third ranked location for Harris Corner detector implementation, while the second ranked of the cv2 function was the top-rank of the Harris implementation. Looking at the comparison of the corner detection methods side by side they both land on the key features, for example the AllmanBrothers it’s the buttons and keys.

Andrew Garcia
7/26/2022

Code

Homework3.py:

```
import cv2

import numpy as np

import math


RED = (0, 0, 255)

GREEN = (0, 255, 0)

BLUE = (255, 0, 0)


def sort_by_max_values(matrix, threshold):

    # Find all values greater than 0

    max_value = np.amax(matrix)

    rows, cols = np.where(matrix > max_value* threshold)


    # Iterate over loc to create list of (r,c,i)

    loc_list = [(r, c, matrix[r,c]) for r,c in zip(rows, cols)]


    # sort by intensity

    sorted_loc = loc_list.copy()

    for i in range(0, len(sorted_loc)):

        for j in range(0, len(sorted_loc)-i-1):

            if (sorted_loc[j][2] < sorted_loc[j + 1][2]):

                temp = sorted_loc[j]

                sorted_loc[j]= sorted_loc[j + 1]

                sorted_loc[j + 1]= temp
    return sorted_loc


def filter_points_by_distance(points_list, distance):
    # Get points that are greater than set distance apart
    for idx1 in range(len(points_list)):
        for idx2 in range(len(points_list)):
            if idx1 == idx2:
                pass
            elif type(points_list[idx1]) == int or type(points_list[idx2]) == int:
                pass
            else:
                r1 = points_list[idx1][0]
                c1 = points_list[idx1][1]
                r2 = points_list[idx2][0]
                c2 = points_list[idx2][1]
                d = math.dist((r1,c1), (r2,c2))
                if d < distance:
                    points_list[idx2] = 0

    # Remove all 0 from list
    try:
        while True:
            points_list.remove(0)
    except ValueError:
        pass
    return points_list
```

Andrew Garcia
7/26/2022

```
def harris_corner_detector(img, maxCorners, alpha, minDistance):
    # Find X and Y gradient components
    ix = cv2.Sobel(img, cv2.CV_32F, 1, 0)
    iy = cv2.Sobel(img, cv2.CV_32F, 0, 1)

    # Find M_bar components by applying gaussian blur to M components
    kernal = (5,5)
    ix_sq_bar = cv2.GaussianBlur(np.multiply(ix,ix), kernal, cv2.BORDER_DEFAULT)
    iy_sq_bar = cv2.GaussianBlur(np.multiply(iy,iy), kernal, cv2.BORDER_DEFAULT)
    ix_iy_bar = cv2.GaussianBlur(np.multiply(ix,iy), kernal, cv2.BORDER_DEFAULT)
    iy_ix_bar = cv2.GaussianBlur(np.multiply(iy,ix), kernal, cv2.BORDER_DEFAULT)
    #
    #M_bar = np.array([[ix_sq_bar, ix_iy_bar],
    #                  [iy_ix_bar, iy_sq_bar]])

    # Formula: Q = det(M_bar) - alpha*(trace(M_bar))^2
    M_bar_det = np.multiply(ix_sq_bar,iy_sq_bar) - np.multiply(ix_iy_bar,iy_ix_bar)
    trace = np.multiply(ix_sq_bar + iy_sq_bar, ix_sq_bar + iy_sq_bar)
    Q = M_bar_det - alpha*trace

    # Get all corner coordinates that are minDistance away
    threshold = 0.5
    sorted_loc = sort_by_max_values(matrix = Q, threshold = threshold)
    corner_coord_ = filter_points_by_distance(points_list = sorted_loc, distance = minDistance)

    # Keep doing this until max Corners is met
    print(f"Loading... finding top {maxCorners} corners")
    while len(corner_coord_) < maxCorners:
        sorted_loc = sort_by_max_values(matrix = Q, threshold = threshold)
        corner_coord_ = filter_points_by_distance(points_list = sorted_loc, distance = 
minDistance)
        threshold = threshold - 0.01
    print(f"Finally got {maxCorners}!!!")


    # Trim down the extras
    corner_coord_ = corner_coord_[:maxCorners]


    # remove intensity from sorted_loc
    corner_coord = [(r,c) for r,c,_ in corner_coord_]
    return corner_coord




def main(filenames):
    for filename in filenames:
        # Read in IMG as color and greyscale
        img1 = cv2.imread(filename)
        img2 = img1.copy()
        img_grey1 = cv2.imread(filename,0)
        img_grey2 = cv2.imread(filename,0)
        # Find 100 corners using CV2
        corners_1 = cv2.goodFeaturesToTrack(img_grey1
                                            , maxCorners = 100
                                            , qualityLevel=0.05
```

```
                                       , minDistance=10)

    # Part 1d: Print to console filename and x,y coord of top 3 corner points
    print("Top 3 corners using cv2.goodFeaturesToTrack")
    print(f'Filename: {filename}')
    for i in corners_1[:3]:
        x = i.ravel()[0]
        y = i.ravel()[1]
        print(f'(x,y): ({x},{y})')


    corners_1 = np.int0(corners_1)
    for i in corners_1:
        x,y = i.ravel()
        cv2.circle(img1, (x,y), 3, GREEN, -1)


    # Show Image
    cv2.imshow(f'CV_corners_{filename}', img1)
    cv2.waitKey(0)
    cv2.imwrite(f'CV_corners_{filename}', img1)




    # Find 100 corners using homemade Harris Corner Detector
    corners_2 = harris_corner_detector(img = img_grey2
                                      , maxCorners = 100
                                      , alpha = 0.05
                                      , minDistance = 10)
```

Andrew Garcia
7/26/2022

```
        # Part 1d: Print to console filename and x,y coord of top 3 corner points

        print("Top 3 corners using Harris Corner Method")

        print(f"Filename: {filename}")

        for i in corners_2[:3]:

            x = i[1]

            y = i[0]

            print(f'(x,y): ({x},{y})')


        for r,c in corners_2:

            cv2.circle(img2, (c,r), 3, RED, -1)


        # Show Image

        cv2.imshow(f'Harris_corners_{filename}', img2)

        cv2.waitKey(0)

        cv2.imwrite(f'Harris_corners_{filename}', img2)


        # Merge the two images

        img_combined = np.concatenate((img1,img2), axis = 1)

        cv2.imshow(f'Combined_{filename}', img_combined)

        cv2.waitKey(0)

        cv2.imwrite(f'combined_{filename}', img_combined)


if __name__ == "__main__":

    filenames = ['AllmanBrothers.png'

             , 'CalvinAndHobbes.png'

             , 'Chartres.png'

             , 'Elvis1956.png']
    main(filenames)
```