

Lab 2 – Harris' corner detector

General information:

- Run files in Anaconda Powershell Prompt e.g. `python example1.py`
- Remember that when you create your own source code, the indentation in loops, whether in conditions should be equal to one tab, or 4 characters
- Follow the instructions in the examples one by one and in the case of the tasks set out, implement them (detailed task is always presented at the end of the instructions), answer the questions and include it in the report. Send the report to the lecturer with the title of e-mail **IPbVSLAB2 RAPORT**

Example 1 - Harris corner detector

- Download a file `chessboard.jpg` from Moodle to the work folder
- Run the script `exampleHAR1.py` available on Moodle's course site
- Rotate the input image (solution suggestion in a yellow frame) and check if the detector determines the intersection of the lines
- Read the definition of the functions `cv2.getRotationMatrix2D` and `cv2.warpAffine`
https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_geometric_transformations/py_geometric_transformations.html

```
import cv2
import numpy as np

filename = 'chessboard.jpg'
img = cv2.imread(filename)
gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

gray = np.float32(gray)
dst = cv2.cornerHarris(gray,2,3,0.04)

#result is dilated for marking the corners, not important
dst = cv2.dilate(dst,None)

# Threshold for an optimal value, it may vary depending on the image.
img[dst>0.01*dst.max()]=[0,0,255]

cv2.imshow('dst',img)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

```
rows,cols = img.shape[1],img.shape[0]
# you can use other parameters
M = cv2.getRotationMatrix2D((cols/2,rows/2), 45, 1);
img = cv2.warpAffine(img,M,(cols,rows))
```

Example 2 – Harris corner detector

- Download a file building.jpg from Moodle to the work folder
- Run the script exampleHAR2.py available on Moodle's course site
- Read the definition of function cornerHarris
https://docs.opencv.org/3.4/dd/d1a/group_imgproc_feature.html#gac1fc3598018010880e370e2f709b4345
- What is a dst result of the function cornerHarris?
- Analyze the influence of the threshold parameter on the number of designated corners

```
from __future__ import print_function
import cv2 as cv
import numpy as np
import argparse

source_window = 'Input image'
corners_window = 'Detected corners'
max_thresh = 255

def demo(val):
    thresh = val

    # detector parameters
    blockSize = 2
    apertureSize = 3
    k = 0.04

    # corners detection
    dst = cv.cornerHarris(src_gray, blockSize, apertureSize, k)

    # Normalization
    dst_norm = np.empty(dst.shape, dtype=np.float32)
    cv.normalize(dst, dst_norm, alpha=0, beta=255, norm_type=cv.NORM_MINMAX)
    dst_norm_scaled = cv.convertScaleAbs(dst_norm)

    # Drawing a crircles around corners
    for i in range(dst_norm.shape[0]):
        for j in range(dst_norm.shape[1]):
            if int(dst_norm[i,j]) > thresh:
                cv.circle(dst_norm_scaled, (j,i), 5, (0), 2)

    # Displaying the results
    cv.namedWindow(corners_window)
    cv.imshow(corners_window, dst_norm_scaled)

# Input image reading
src = cv.imread('building.jpg')
if src is None:
    print('Nie można wczytać obrazka:', args.input)
    exit(0)
src_gray = cv.cvtColor(src, cv.COLOR_BGR2GRAY)

# Creating the window with trackbar
cv.namedWindow(source_window)
thresh = 200 # initial value of threshold
cv.createTrackbar('Prog: ', source_window, thresh, max_thresh, demo)
cv.imshow(source_window, src)
demo(thresh)
cv.waitKey()
```

Example 3 - Harris corner detector, good corners.

- Download a file exampleHAR3.py from Moodle to the work folder
- Read the definition of cv2.goodFeaturesToTrack function and its parameters
- https://docs.opencv.org/2.4/modules/imgproc/doc/feature_detection.html#goodfeaturestotrack

```
import cv2
import numpy as np

img = cv2.imread('building.jpg', cv2.IMREAD_GRAYSCALE)

corners = cv2.goodFeaturesToTrack(img, 20, 0.01, 10)
corners = np.int0(corners)

for i in corners:
    x, y = i.ravel()
    cv2.circle(img, (x, y), 3, 255, -1)

cv2.imshow('good corners', img)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

Homework

- Prepare the program code whose purpose will be to mark the 4 corners of the page recorded on the camera input image. The result of the program should be a window with the image of the online camera, with a card presented in front of the camera in motion and marked corners in this image.
- Reporting on the implementation of the task:
 - program code in the .py file to run
 - Answers to the questions contained in this manual, plus any hints as to the conditions of the experiment if they are quite specific when running the uploaded, written by you code should be in a separate PDF file.