Maak Ali(2k18/IT/56),

Farzeen Hassan(2K18/IT/39)

Report

Introduction: This is computer vision and image processing group assignment. I have taken code from slides provided by sir and then tried to modify code and see the results. I have applied canny edge detector to detect edges, Hough transform to detect HT lines and Harris corner detector to detect corners of the image.

Problem

* True Edges of the pictures to be detected.
* Hough transforms lines to be detected.
* True Corners of the pictures to be detected.

Solution

* We are using canny edge detector to detect our true edges.
* We are using Hough transform for detecting the HT lines of the pictures.
* We are using Harris corner detector to detect the corners.

Modification

* This code is collected from the slides. This project is the combination multiple programs given in slides.
* We have used sk image library to save results in disk to understand our results without executing code. **skimage**, is an open source Python package designed for image preprocessing.
* We have changed the value of threshold 1 and threshold 2 to get efficient results according to our image.
* We have saved our result in disk so that we can understand the result without executing code.
* We have changed kernel value of Harris corner detector to get understandable results.
* We have saved and showed the results of dilated and undilated image to understand the effect of dilation on our corner detection.

Algorithm

* We have read the image from disk.
* We have changed image to gray scale.
* We have detected the edges by canny edge detector.
* We have detected HT lines.
* We have showed and save the result of edge detection and HT line detection.
* We have detected the corners of image using Harris corner detector.
* We have showed and saved the results of the corner detection.

Conclusion: we got our desired results by applying canny edge detector, Harris corner detector and Hough transform on our image.

Code

import cv2

import numpy as np

from skimage import io

# read image from disk

myImage = cv2.imread('Maak.png')

# convert BGR to gray scale

gray = cv2.cvtColor(myImage, cv2.COLOR\_BGR2GRAY)

# find edges using canny edge detector

cannyEdges = cv2.Canny(gray, 100, 200)

# find lines in image

lines = cv2.HoughLines(cannyEdges, 1, np.pi / 180, 200)

for line in lines:

rho, theta = line[0]

a = np.cos(theta)

b = np.sin(theta)

x0 = a \* rho

y0 = b \* rho

x1 = int(x0 + 1000 \* (-b))

y1 = int(y0 + 1000 \* (a))

x2 = int(x0 - 1000 \* (-b))

y2 = int(y0 - 1000 \* (a))

cv2.line(myImage, (x1, y1), (x2, y2), (0, 0, 255,), 2)

# show result

cv2.imshow('Canny Edges', cannyEdges)

cv2.imshow('HT lines', myImage)

# save result

io.imsave('result/Canny edges.png', cannyEdges)

io.imsave('result/HT lines.png', myImage)

# harris Corner detector

gray = np.float32(gray)

harrisCorner = cv2.cornerHarris(gray, 4, 7, 0, 0.04)

# result is dilated for marking the corners,not important

dilated = cv2.dilate(harrisCorner, None)

# Threshold for an optimal value, it may vary depending on the image.

myImage[dilated > 0.01 \* dilated.max()] = [0, 0, 255]

# show result

cv2.imshow('Hough Transform lines & harris corners', myImage)

cv2.imshow('Harris Corners (With out Dilated)', harrisCorner)

# save result in disk

io.imsave('result/Harris Corner detection & HT lines.png', myImage)

io.imsave('result/Harris Corners (With out Dilated).png', harrisCorner)

# waiting for key

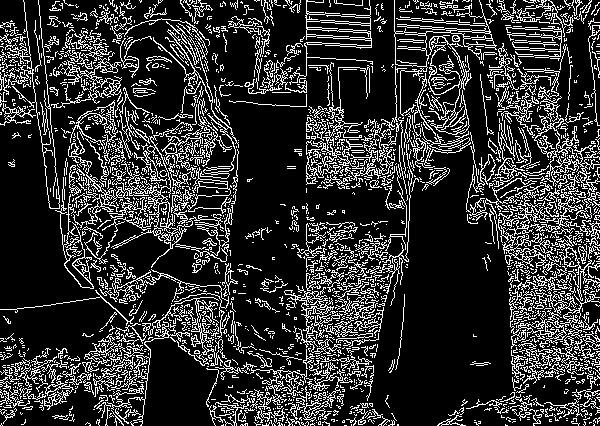
k = cv2.waitKey(0)

if cv2.waitKey(0) & 0xff == 27:

cv2.destroyAllWindows()

Results:

Harris Corner Detection and HT lines Canny Edges



HT lines Harris Corner(without dilated)

