

Project 01: Applications of Line and Circle Hough Accumulator Computational Perception and Artificial Intelligence

Description:

In Problem Set 06 we defined functions to create Hough Accumulators for Lines and Circles. Now it is time to apply line and circle detection with some real world intelligence tasks:

- A. Identifying Regular Polygons
- B. Counting Change
- C. Analyzing Artwork

In this project you will use the Hough Algorithms from PS06 to complete the above tasks. You will need to be creative in how you apply the algorithms to complete the task. You will not be provided with step by step instructions. Instead, you will implement functions to complete the tasks listed below and output results as images.

Finally, for tasks A and B, you will use the camera instead of pre-loaded pictures. The idea is to build a system that can analyze images in (almost . . .) real time. Good luck and don't be afraid to "Fail Fast!"

File Setup:

1. Create a Python project in PyCharm Named "Project01"
2. Create a folder "input"
3. Create a folder "output"
4. Create a Python module called "util.py" and copy the following functions from PS06
 - a. `normalize(img)`
 - b. `houghLineAccumulator(E)`
 - c. `getHoughVectors(H)`
 - d. `drawLines(img, V)`
 - e. `houghCircleAccumulator(E, r)`
 - f. `getHoughCircles(H, r)`
 - g. `drawCircles(img, C, color)`

Note: use `from util import *` to access these functions in your other python modules

Task A: Identifying Regular Polygons

Materials Needed:

- a. Printout or block shapes of triangle, square, pentagon, hexagon, octagon, and star with shape in black.
 - a. You may print out this document: <http://nebomusic.net/perception/Shapes.pdf>
- b. White table surface for contrast.
- c. Webcam and Ring stand

Requirements:

1. Create a python module called `taskA.py` to and write code in this module to complete task.
2. Capture an image from the camera of shape on surface (Code sample below)

```
# Sample Camera Read and Save for Project01
# Mr. Michaud
# www.nebomusic.net

import cv2 as cv2
import numpy as np

# Setup Camera Object: 0 is the index of the installed camera
# 1 is the index of a USB camera (webcam)
cap = cv2.VideoCapture(0)

# Capture One Frame
ret, img = cap.read()

# Save img if needed.
cv2.imwrite("output/cameraCapture.jpg", img)
```

3. Identify the number of sides on the shape and thus the type of shape. This should be rotationally invariant (meaning you can count lines in any orientation)
4. The code should draw the correct lines and print the number of lines and type of shape on the screen.
 - a. Use `cv2.putText(img, textString, (x, y), size, color)`
5. Use the camera and code to identify and create output images for each shape. Save in output folder.
 - a. `triangle.jpg`
 - b. `square.jpg`
 - c. `pentagon.jpg`
 - d. `hexagon.jpg`
 - e. `octagon.jpg`

Task B: Counting Change

Materials Needed:

- a. Collection of Coins (pennies, nickels, dimes, quarters)
- b. Black surface
- c. Webcam and Ring stand

Requirements:

1. Create a python module called `taskB.py` and write code in this module to complete task.
2. Capture an image using the Camera, ring stand, and coins. Code sample is provided with Task A requirements.
3. Identify the type and number of coins in the images. Use the Hough Circle functions you wrote and identify the radius of the coins (make sure you consistently setup the ring stand to get the same results each day.). Once you can count the amount of each type of coin, add up the change and store the value of all the coins in the image.
4. Draw the appropriate sized circles around the coins and use the `cv2.putText()` function to print the value of the coins added together on the image.
5. Run at least 4 different configurations of change and save as the file names below in the output folder.
 - a. `changeSample1.jpg`
 - b. `changeSample2.jpg`
 - c. `changeSample3.jpg`
 - d. `changeSample4.jpg`

Task C: Analyzing Artwork

Materials Needed:

- a. Phone Camera or other digital Camera
- b. Access to artwork in Gunn Building

Requirements:

1. Create a python module called `taskC.py` and write code in this module to complete task.
2. Take at least two pictures with your phone or other camera featuring two different pieces of art on display in the Gunn Building Art Gallery (use student artwork). Save these images in your 'input' folder as `art1.jpg` and `art2.jpg`
3. Using the Hough Line and Circle accumulators, identify the key lines and circles on the artwork images. Work to get the best match of clean lines and circles. Save these images in the output folder as:
 - a. `houghArt1.jpg`
 - b. `houghArt2.jpg`
4. Show the Hough analysis images from step 3 to Mr. DeAngelis or to the students who created the art. Gather the reaction from Mr. DeAngelis or the artists on the role of circles and lines in their artwork and whether the Hough Algorithms captured the essential lines and circles. Ask them how the algorithm can improve in capturing the "essence of visual art". (Be prepared for any kind of answer – even skeptical).
5. Do one of the following:
 - a. Write a paragraph describing and quoting the reactions from Mr. DeAngelis or the student artists. Save this paragraph in .docx format in the output folder.
 - b. Capture a video of Mr. DeAngelis or the student artists answering the questions and reacting to the Hough Images from step 4. Post these videos on YouTube as private videos and include the links in the comments of `taskC.py`.