STB600 Lab 5: Features extraction

Atieh Sahraeidolatkhaneh: atieh.sahraeidolatkhaneh@hv.se

Yongcui Mi: yongcui.mi@hv.se

1. Introduction

Image feature extraction is a process to construct derived values (features) that are intended to be informative and non-redundant. Image features include colors, gray levels, shapes, textures and more broadly any piece of information which is relevant for solving a certain application. When dealing with image classification, features should be most relevant for discrimination and have low intra-class variability and high between-class variability. There are two principal aspects of image feature extraction: feature detection, and feature description. Numerical feature descriptions usually are "packaged" in the form of a feature vector, (i.e., a 1 $\times$ n or n $\times$ 1 matrix) and be further processed.

The shape of an object is an important and basic visual feature for describing image content, therefore, the first task of this lab is to learn how to extract different object shape features, such as areas, perimeters and etc. This is mainly done by extracting contour from images and then calculate the contour properties. Then introduce the scale-invariant feature transform (SIFT) algorithm as the second task.

(PS: You can go through the functions using pictures from lab 4. If you like to explore more, for instance to classify images by extracted features, you will find two subfolders containing two different classes of images from this lab folder.)

2. Tasks: Calculate different contour features

To find the different features of contours, like area, perimeter, centroid, bounding box etc. First find the contours using cv.findContours(), then get the properties of the contour by using different functions as shown below:

```

#Moments: M = cv.moments(cnt)
#Area: area = cv.contourArea(cnt)
#Perimeter: perimeter = cv.arcLength(cnt,True)
#Convex Hull: hull = cv.convexHull(cnt)
#Checking Convexity: k = cv.isContourConvex(cnt)
```

```
#Straight Bounding Rectangle:
    x,y,w,h = cv.boundingRect(cnt)
    cv.rectangle(img,(x,y),(x+w,y+h),(0,255,0),2)
# Rotated Rectangle:
    rect = cv.minAreaRect(cnt)
    box = cv.boxPoints(rect)
    box = np.int0(box)
    cv.drawContours(img,[box],0,(0,0,255),2)
#Fitting an Ellipse
    ellipse = cv.fitEllipse(cnt)
    cv.ellipse(img,ellipse,(0,255,0),2)
#Orientation:
(x,y),(MA,ma),angle = cv.fitEllipse(cnt)
# Aspect Ratio:
    x,y,w,h = cv.boundingRect(cnt)
    aspectRatio = float(w)/h
#Equivalent Diameter:
    area = cv.contourArea(cnt)
    equi_diameter = np.sqrt(4*area/np.pi)
# Extreme Points:
    leftmost = tuple(cnt[cnt[:,:,0].argmin()][0])
    rightmost = tuple(cnt[cnt[:,:,0].argmax()][0])
    topmost = tuple(cnt[cnt[:,:,1].argmin()][0])
    bottommost = tuple(cnt[cnt[:,:,1].argmax()][0])
```

3. An Introduction to Scale Invariant Transform Algorithm (SIFT)

The scale-invariant feature transform (SIFT) is a computer vision algorithm to detect, describe, and match local features in images. The objective is to find the keypoints in the images. Try to run the algorithm and understand how it works.

First, construct a SIFT object, then detect the keypoints and draw them on the image, you can also search only a specific part of the image. Try to understand the algorithm and explain the reason behind extracting the keypoints in images.Explain what the inputs for the 'detect' function are. Use the following scripts:

```
# In case of error regarding missing a module
 try to insatall 'opencv-contrib-python' in your system or working
   directory
```

```
3
4 # Sift Object: cv.SIFT_create()
5 # Keypoints:  sift.detect(grayscale img,None)
6 # draw points: cv.drawKeypoints(grayscale img,points,img)
```

4. Additional resource

References