CSCI435/CSCI935 Computer Vision: Algorithms and Systems

Spring 2023

Assignment Two (20%)

Due Date: On Moodle

Objectives

- Design a Python program using OpenCV that detects keypoints from an image and compare two images using **SIFT detector and descriptors**.

Research has revealed that keypoint-based descriptors are effective to characterize both individual objects and entire images. Such descriptors are widely used in object detection and image comparison. In this assignment, you are required to develop a program in Python using OpenCV 4.6.0 to detect and display keypoints from an image and to compare images using **SIFT detector and descriptors**. Parameters to the SIFT detector and feature extractor need to be tuned to the provided set of images.

The program should be able to take **one** or **multiple image files**.

Task One

When a single image file, e.g. `A.jpg`, *is supplied to the program*, it should

1. Rescale properly the image to a size **comparable** to VGA size (480(rows) x 600(columns)) to reduce the computation. Note that the aspect ratio of the image should be kept when the image is rescaled.
2. Extract **SIFT** keypoints from the luminance Y component of the image
3. For each detected keypoint, draw a **cross** "+" at the location of the key point and **a circle** around the keypoint whose radius is proportional to the scale of the keypoint and **a line** from the "+" to the circle indicating the orientation of the key point.
4. Both the original image and the image with highlighted keypoints should be displayed in a window as follow

Original Image (rescaled)	**Image with highlighted keypoints**

5. In a command window or Windows Powershell, the program should output the number of detected keypoints, e.g.

```
# of keypoints in A.jpg is 3180
```

Task Two

When multiple image files, say `A.jpg, B.jpg, C.jpg, D.jpg and E.jpg`, are supplied to the program via command arguments, the program will compare each pair of the images using a Bag-of-Words model constructed from SIFT descriptors. Specifically, the program shall

1. Rescale properly all images to sizes comparable to VGA size (480x600) to reduce computation. Note that the aspect ratio of the images should be kept when the image is rescaled, but there is no need to rescale all images to the same size.
2. Extract SIFT keypoints and descriptors from the luminance Y component of all scaled images.
3. Cluster the SIFT descriptors extracted from ALL images into K-clusters using K-means algorithm. The values of K should be specified as a percentage of the total number of keypoints. Each cluster represents a visual word.
4. For each image, construct a histogram of the occurrence of the visual words. This should be done by classifying each SIFT descriptor of the image into one of the K-clusters (words) and continuing how many time each word occurred in the image.
5. For each pair of images, calculate the χ^2 distance between the normalized histograms of visual words of the images. This χ^2 distance is a measurement of the ***dissimilarity*** of the pair of images.
6. The program should **output** the following information
    a. Number of keypoints for each image and the total number of keypoints of all images, e.g.

    ```
    # of keypoints in A.jpg is 2138
    # of keypoints in B.jpg is 923
    # of keypoints in C.jpg is 780
    # of keypoints in D.jpg is 1300
    # of keypoints in E.jpg is 1578
    .........
    ```

    b. Dissimilarity matrices for K=5%, 10% and 20% of the total number of keypoints from all images. Note: please arrange the dissimilarity matrices in a readable format, e.g.

    ```
    K=5%*(total number of keypoionts)=250

    Dissimilarity Matrix

          A     B     C     D     E
     A    0.0   0.01              0.9
     B          0.0
     C                0.0
     D                      0.0   0.85
     E                            0.0


    K=10%*(total number of keypoionts)=500

    Dissimilarity Matrix
    ```

```
      A     B      C     D     E
 A    0.0   0.01               0.9
 B          0.0
 C                 0.0
 D                       0.0   0.85
 E                             0.0


K=20%*(total number of keypoionts)=1000

Dissimilarity Matrix

      A     B      C     D     E
 A    0.0   0.01               0.9
 B          0.0
 C                 0.0
 D                       0.0   0.85
 E                             0.0
```

However, the program neither needs to display the original images nor the images with highlighted keypoints in *Task two*.

Note that fifteen images covering different scenarios are provided for testing your program. Your program should work for any image (Task one) and any number of images (Task two)

Requirements on implementation

1. The program should be named as "**siftImages**" and shall take one or multiple image files as input, e.g. **`python`** `siftImages.py` *`imagefile1 [imagefile2 imagefile3 imagefile4 …]`*.
2. No other third-party libraries should be used in the program except OpenCV 4.6.0-Python (assuming that `numpy` and `matplotlib` packages exist). The code must be in Python.
3. SIFT is an extra-module in OpenCV. You must install OpenCV-Python using "`pip install opencv-contrib-python`" in order to include extra modules.
4. The code should be modularized with detailed comments AND all source code should be placed in a single file "`siftImages.py`".

Marking Scheme

1. Zero marks may be graded if your code cannot run as specified in the requirements.
2. Program structure, comments and usability (1%)
3. Proper rescale of input image(s) (2%)
4. Extraction of SIFT keypoints (2%)
5. Display of the original image and image with highlighted SIFT keypoints (5%)
6. Extraction of SIFT keypoints of the multiple images and generate the visual words (3%)
7. Calculate and generate the dissimilar matrices (7%)

Submission

Zip the `siftImages.py` file to ***your_login_name*.zip**. The zip file must be submitted via Moodle.

IMPORTANT:

a) ***DO NOT*** *include and submit any object files and images in the zip file. Your submission may not be accepted if you do so.*

b) Submission through email ***WILL NOT*** be accepted