

CSCI435/CSCI935 Computer Vision: Algorithms and Systems

Spring 2018

Assignment Two (20%)

Due Date: 11:59pm 23 September 2019

Objectives

- Design a C (or C++) program that detects keypoints from an image and compare two images based on SIFT descriptors.
- The assignment can be completed by either **individuals** or **a group of two students**.

Task

Recent 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 C/C++ using OpenCV 3.4.1 to detect and display keypoints from an image and to compare two images based on SIFT descriptors.

The program should be able to take **one** or **multiple image files**. When a single image file, e.g. A.jpg, is supplied to the program, it should

1. Rescale 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 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.
4. Both the original image and the image with highlighted keypoints should be displayed in a window as follow



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

of keypoints in A.jpg is 3180

When multiple image files, say A.jpg, B.jpg, C.jpg, D.jpg and E.jpg, are supplied to the program as 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 all images to sizes comparable to VGA size (480x600) to reduce the 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 Y-components of all scaled images.
3. Cluster the SIFT descriptors 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 histograms 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 A.jpg is 780
# of keypoints in B.jpg is 1300
# of keypoints in A.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 keypoints)=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 keypoints)=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 keypoints)=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 this case.

Fifteen images covering different scenarios are provided for testing your program.

Requirements on coding

1. The program should be named as “**siftImages**” and shall take one or multiple image files as input, e.g. `siftImages imagefile1 [, imagefile2, imagefile3,...]`.
2. No other third-party libraries should be used in the program except OpenCV 3.4.1. The code has to be in C/C++.
3. The code should be modularized with detail comments AND all source code should be placed in a single file “`siftImages.cpp`” or “`siftImages.c`”.

Marking Scheme

1. Zero marks may be graded if your code cannot be compiled using *MS VS 2015 and openCV 3.4.1*.
2. Program structure, comments and usability (2%)
Specific comments on how you rescale the size of the input image(s) must be included in the beginning of your source C/C++ file
3. Proper rescale of input image(s) (1%)
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

1. Zip the SOURCE file to *your_login_name.zip*. The zip file has to be submitted in Moodle.
2. If the assignment is completed by a group of two students.
 - a) Only one copy of the assignment should be submitted by one of the students.
 - b) At beginning of the source file, specify both students' login names and student ids in the comments

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