COMP9517: Computer Vision

2021 T3 Lab 2 Specification

Maximum Marks Achievable: 2.5

This lab is **worth 2.5% of the total course marks**.

The lab files should be submitted online.
Instructions for submission will be posted closer to the deadline.
Deadline for submission is Week 4, Wednesday 6 October 2021, 23:59:59.

Objective: This lab revisits important concepts covered in the lectures of Week 3 and aims to make you familiar with implementing specific algorithms.

Materials: The sample image as well as template code to be used in the tasks of this lab is available in WebCMS3. You are required to use OpenCV 3+ with Python 3+.

Submission: The tasks are assessable **after the lab**. Submit your source code as a Jupyter notebook (.ipynb) with output images (.png) in a single zip file via Moodle by the above deadline. The submission link will be announced in due time.

The sample image **Beehive_Corner_Adelaide.png** is to be used for all questions.

SIFT: Scale-Invariant Feature Transform

A well-known algorithm in computer vision to detect and describe local features in images is the scale-invariant feature transform (SIFT). Its applications include object recognition, mapping and navigation, image stitching, 3D modelling, object tracking, and others.

A SIFT feature of an image is a salient keypoint with an associated descriptor. SIFT computation is commonly divided into two steps:

1) detection,
2) description.

At the end of the detection step, for each keypoint the SIFT algorithm computes the:

- keypoint spatial coordinates (x, y), 坐标
- keypoint scale (in the scale space), 比例
- keypoint dominant orientation. 主导方向

The subsequent description step computes a distinctive 128-dimensional feature vector for each keypoint. SIFT is designed in such a way that this descriptive feature vector is invariant to scaling and rotation. Moreover, the algorithm offers decent robustness to noise, illumination gradients, and affine transformations.

SIFT in OpenCV

In OpenCV the SIFT algorithm is available only in the non-free module (OpenCV has both free and non-free modules). The algorithm has been patented by the creator but can be freely used for academic and research purposes. The non-free modules can be found in the opencv_contrib package. You will first need to install this package, as shown below, and then you can use the SIFT module.

Initialize and activate the virtual environment (optional):

```
$ python3 -m venv env
$ source venv/bin/activate
```

Install the correct version of OpenCV and the contrib module:

```
$ pip install opencv-python==3.4.2.17
$ pip install opencv-contrib-python==3.4.2.17
```

Task 1 (0.5 mark): Compute the SIFT features of the given image.

a) Extract SIFT features with default parameters and show the keypoints on the image.
b) To achieve better visualization of the keypoints, reduce the number of keypoints.
   Hint: Vary the parameter **contrastThreshold** or **nfeatures** so that the number of keypoints becomes about 10% of all default keypoints.

Submit the images obtained in a) and b) and briefly describe in your Jupyter notebook the approach you used for b).

Task 2 (1 mark): Change the scale of the image and recompute the SIFT features.

a) Enlarge the given image by a scale percentage of 120.
b) Extract the SIFT features and show the keypoints on the scaled image using the same parameter setting as for Task 1 (for the reduced number of keypoints).
c) Inspect the keypoints visually: Are the keypoints of the scaled image roughly the same as those of the original image? What does this observation imply?
d) Match the SIFT descriptors of the keypoints of the scaled image with those of the original image using the nearest-neighbour distance ratio method. Show the keypoints of the 5 best-matching descriptors on both the original and the scaled image.
   Hint: Brute-force matching is available in OpenCV for feature matching.

Submit the images obtained in b) and d) and include in your Jupyter notebook your answers to the questions in c).

Task 3 (1 mark): Rotate the image and recompute the SIFT features.

a) Rotate the given image anticlockwise by 60 degrees.
b) Extract the SIFT features and show the keypoints on the rotated image using the same parameter setting as for Task 1 (for the reduced number of keypoints).
c) Inspect the keypoints visually: Are the keypoints of the rotated image roughly the same as those of the original image? What does this observation imply?
d) Match the SIFT descriptors of the keypoints of the rotated image with those of the original image using the nearest-neighbour distance ratio method. Show the keypoints of the 5 best-matching descriptors on both the original and the rotated image.

Submit the images obtained in b) and d) and include in your Jupyter notebook your answers to the questions in c).

Coding Requirements and Suggestions

Check OpenCV documentation to find various in-built functions to find SIFT features, draw key points, and matching key points in images. You should understand how the algorithm works, what parameters you can set in these in-built functions, and how these parameters affect the output. For your reference, below are links to relevant OpenCV functions.

2D Features Framework
https://docs.opencv.org/4.5.3/da/d9b/group__features2d.html

Drawing Functions of Keypoints and Matches
https://docs.opencv.org/4.5.3/d4/d5d/group__features2d__draw.html

Descriptor Matchers
https://docs.opencv.org/4.5.3/d8/d9b/group__features2d__match.html

OpenCV SIFT Class reference
https://docs.opencv.org/4.5.3/d7/d60/classcv_1_1SIFT.html

In your Jupyter notebook, the input image should be readable from the location specified as an argument, and all output images and other requested results should be displayed in the notebook environment. All cells in your notebook should have been executed so that the tutor/marker does not have to execute the notebook again to see the results.

Refer to the link below to understand image features and various feature detectors:
https://docs.opencv.org/4.5.3/db/d27/tutorial_py_table_of_contents_feature2d.html.

Also, refer to https://docs.opencv.org/3.4.3/da/df5/tutorial_py_sift_intro.html for an example of computing SIFT features and showing the key points on the image.

And finally see https://docs.opencv.org/4.5.3/dc/dc3/tutorial_py_matcher.html for an example of feature matching.

A template is provided in WebCMS for this lab.

Reference: D. G. Lowe. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, vol. 60, no. 2, pp. 91-110, November 2004. https://doi.org/10.1023/B:VISI.0000029664.99615.94

Released: 28 September 2021