COMP9517

Lab 2, T3 2019 (04/10/2019)

The tasks presented in this document provide revision of important concepts from week 3 lectures (feature representation). You can use OpenCV or other packages for this lab.

NOTE if you are using OpenCV: We will be exploring the SIFT (Scale Invariant Feature Transform) algorithm, which is only available in OpenCV's non-free module (OpenCV has both free and non-free modules). This algorithm has been patented by the creator but is free to use for academic and research purposes. The non-free modules can be found in the opencv_contrib package. You will need to install this package as shown below and then you can use the SIFT module.

Installation:

```
Initialize and activate virtual environment (optional):

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

Install correct version of OpenCV and contrib module:

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

Result images should be submitted in a zip file via WebCMS3 for marking (1 mark). Submission is due at 23:59:59 on Oct 4th, 2019. Submission instruction will be posted prior to the lab session.

The sample image "Eiffel_Tower.jpg" is to be used for task 1 and task 2. And the sample image "road_sign.jpg" is to be used for task 3.

SIFT (Scale Invariant Feature Transform)

SIFT is a well-known algorithm in computer vision to detect and describe local features in images. Its applications include object recognition, robotic mapping and navigation, image stitching, 3D modelling, video tracking and others.

A SIFT feature is a salient keypoint that corresponds to an image region and has an associated descriptor. SIFT computation is commonly divided into two steps:

- detection
- description

At the end of the detection step, and for each feature detected, the SIFT algorithm establishes:

- keypoint spatial coordinates (x, y)

- keypoint scale
- keypoint dominant orientation

After the detection step, the description step computes a distinctive fingerprint of 128 dimensions for each feature. The description obtained is designed to be invariant to scale and rotation. Moreover, the algorithm offers decent robustness to noise, illumination gradients and affine transformations.

Lab Tasks

Task 1: Compute SIFT features:

- Extract SIFT features with default parameters and show the keypoints on the image.
- Reduce the number of keypoints extracted so that the visualisation of keypoints is easier. Show these keypoints on the image. (Hint: vary the parameter "contrastThreshold" or "nfeatures" so that the number of keypoints becomes about ¼ of all default keypoints).

Task 2: Rotate the image and compute the SIFT features again:

- Rotate the image clockwise by 60 degrees
- Extract SIFT features and show the keypoints on the rotated image, using the same parameter settings as task 1 (for reduced number of keypoints)
- Inspect the keypoints visually: Do the keypoints look roughly the same as those extracted on the original image? What does this observation imply?

Task 3: Test rotational invariance of SIFT features:

- Rotate the sample image in increments of 45 degrees, from 0 to 90 degrees.
- For each rotated image, compute SIFT features and draw keypoints on the image.
- For each rotated image, match its SIFT descriptors with those from the original image based on the nearest neighbour distance ratio method.
- Draw the matches between the two images.

Submit 3 images showing the keypoints matches with different rotation degrees (0 degrees, 45 degrees and 90 degrees) in tasks 3 in a zip file for marking.

NOTE: 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 keypoints. And refer to https://docs.opencv.org/trunk/dc/dc3/tutorial_py_matcher.html for an example of feature matching. The attached template.py file can be used as a template for the lab tasks.

REFERENCES

D. G. Lowe, "Distinctive image features from scale-invariant keypoints," Int. J. Comput. Vis., vol. 60, no. 2, pp. 91–110, 2004