National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Department of Software Engineering

CS 474: Computer Vision

Class: BESE-7

Lab 4: Local Image Features

Date: 10th Feb 2020

Time: 10:00 am-1:00 pm

Instructor: Dr. Muhammad Moazam Fraz

Lab Engineer: Ms Anum Asif

Course Learning Outcomes (CLOs)

Upon completion of the course, students should demonstrate the ability to:		**PLO Mapping****	**BT Level***
CLO 1	Understand computer vision algorithms and tools and techniques.	PLO 1	C2
CLO 2	Develop solutions for image/video understanding and recognition.	PLO 3	C3
CLO 3	Use modern tools to solve practical problems.	PLO 5	C5

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

- *Knowledge(C-1), Comprehension(C-2), Application(C-3), Analysis(C-4), Synthesis(C-5), Evaluation(C-6)*

*** PLOs are published on department website*

Lab 4 : Local Image Features

Learning Outcome

CLO 1: Understand computer vision algorithms and tools and techniques.

Tools/Software Requirement

Python / MATLAB

Local Image Features

Local features and their descriptors, which are a compact vector representations of a local neighborhood, are the building blocks of many computer vision algorithms. Their applications include image registration, object detection and classification, tracking, and motion estimation. Using local features enables these algorithms to better handle scale changes, rotation, and occlusion.

Algorithms for detection Corner Features

- FAST
- Harris
- Shi & Tomasi

Algorithms for detection Blob Features

- SURF (Speeded Up Robust feature)
- MSER (maximally stable extremal regions)

Descriptors

- SURF
- FREAK
- BRISK
- HOG

You can mix and match the detectors and the descriptors depending on the requirements of your application.

What is blob detection?

In computer vision, blob detection methods are aimed at detecting regions in a digital image that differ in properties, such as brightness or color, compared to surrounding regions. Informally, a blob is a region of an image in which some properties are constant or approximately constant; all the points in a blob can be considered in some sense to be similar to each other.

What Are Local Features?

Local features refer to a pattern or distinct structure found in an image, such as a point, edge, or small image patch. They are usually associated with an image patch that differs from its immediate surroundings by texture, color, or intensity. What the feature actually represents does not matter, just that it is distinct from its surroundings. Examples of local features are blobs, corners, and edge pixels.

What Makes a Good Local Feature?

Detectors that rely on gradient-based and intensity variation approaches detect good local features. These features include edges, blobs, and regions. Good local features exhibit the following properties:

- **Repeatable detections**

  When given two images of the same scene, most features that the detector finds in both images are the same. The features are robust to changes in viewing conditions and noise.
- **Distinctive**
  The neighbourhood around the feature centre varies enough to allow for a reliable comparison between the features.
- **Localizable**:
  The feature has a unique location assigned to it. Changes in viewing conditions do not affect its location.

Feature Detection and Feature Extraction

Feature detection selects regions of an image that have unique content, such as corners or blobs. Use feature detection to find points of interest that you can use for further processing. These points do not necessarily correspond to physical structures, such as the corners of a table. The key to feature detection is to find features that remain locally invariant so that you can detect them even in the presence of rotation or scale change.

Feature extraction involves computing a descriptor, which is typically done on regions centered around detected features. Descriptors rely on image processing to transform a local pixel neighborhood into a compact vector representation. This new representation permits comparison between neighborhoods regardless of changes in scale or orientation.

Descriptors, such as SIFT or SURF, rely on local gradient computations.

Binary descriptors, such as BRISK or FREAK, rely on pairs of local intensity differences, which are then encoded into a binary vector.

Choose a Feature Detector and Descriptor

Select the best feature detector and descriptor by considering the criteria of your application and the nature of your data. The table below helps you understand the general criteria to drive your selection.

Considerations for Selecting a Detector and Descriptor

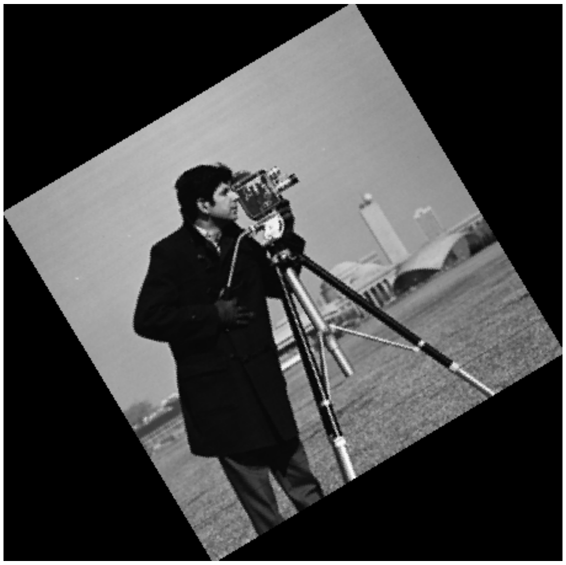
Criteria	Suggestion
Type of features in your image	Use a detector appropriate for your data. For example, if your image contains an image of bacteria cells, use the blob detector rather than the corner detector. If your image is an aerial view of a city, you can use the corner detector to find man-made structures.
Context in which you are using the features: Matching key points Classification	The HOG and SURF descriptors are suitable for classification tasks. In contrast, binary descriptors, such as BRISK and FREAK, are typically used for finding point correspondences between images, which are used for registration.
Type of distortion present in your image	Choose a detector and descriptor that addresses the distortion in your data. For example, if there is no scale change present, consider a corner detector that does not handle scale. If your data contains a higher level of distortion, such as scale and rotation, then use the more computationally intensive SURF feature detector and descriptor.
Performance requirements: • Real-time performance required • Accuracy versus speed	Binary descriptors are generally faster but less accurate than gradient-based descriptors. For greater accuracy, use several detectors and descriptors at the same time.

Lab Task

Experiment with the Local Feature Detectors

- Load an image and save its copy as a geometrically transformed image (i.e. rotated image).

Original Image



Distorted Image with Rotation

- Apply local feature detection / extraction to visualize the repeatability of different features detectors. You can use any image for your experiments.
- You can use following feature detectors to identify key points/local features, draw the keypoint locations on the geometrically transformed image and observe their repeatability.

```
cv2.cornerHarris
cv2.FastFeatureDetector()
cv2.ORB()
```

Deliverable

- Jupyter Notebook submitted on LMS