The image retrieval algorithm is divided into two parts: Offline computation and Online computation.

1. Offline computation

In the Offline stages, we perform number of computation for the entire image database. The steps are performed in this steps are:

1.1 Detector computation

In this module, different state of the art detectors(e.g. harris, hesaian-affine, mser etc.) will be computed for the entire image dataset.

1.2.1 Descriptor computation

The extracted detector in the above steps will be described using several state of the art descriptors(e.g. sift, surf etc.). While computing different descriptors, there are possibilities to mismatch in number of description points for different descriptor.

1.2.2 Label Matrix generation

After extracting descriptors, all the extracted descriptions points will be merged in a single file called LabelMatrix where we have information about each description points vectors and associated the image id.

1.3.1 Codebook generation

For each descriptors, we compute different size of CodeBook, e.g. 50000, 100000 etc using previously generated LabelMatrix. In general approximately 30% of the total description points are chosen as CodeBook size, i.e. if the total description points in LabelMatrix is 160000, the size of CodeBook can be 50000.

1.3.2 KdTree computation

Kd Tree is computed for nearest neighbor search.

1.4 Inverted unique index generation

Different inverted index files are generated for different codeBook size.

2. Online computation for query images

For a query image, we perform below mentioned steps during online computation:

2.1 Detector computation

2.2.1 Descriptor computation

2.2.2 Label Matrix generation

2.5 Distances min-max computation

2.6 Image Retrieval

2.7 Output

The entire workflow is organized as described below:

0_ImgDatasets → Contains image datasets

In the image dataset, we organized the entire dataset in different image classes. The image dataset contains different folder of image classes, i.e. 01, 02...etc classes. Inside each folder, several images belong to that particular class are present.

1_Detectors → Contains computed detectors:

data – DatasetName – detector

2_Descriptors → Contains computed descriptors:

data – DatasetName – detector+descriptor

3_CodeBooks → Contains CodeBooks:

data – DatasetName – detector+descriptor

4_InvertedUniqueIndices → Contains inverted index

data – DatasetName – detector+descriptor

5_ComputeDistances

data – DatasetName – detector+descriptor – SizeOfCodeBook

6_ImageRetrieval

Results

bins → Contains all bins files

Each folder contains several scripts (shell, python) are present to perform different steps in the algorithm. Most of the codes are written in C++(using OpenCV library) and few parts are written in Python and Shell Scripts.

There are two scripts:

OfflineScript.sh → Contains all the Offline steps computation scripts.

OnlineScript.sh → Contains all the Online steps computation scripts for query image.

- First, we run OfflineScript by entering image dataset name, detectors and descriptors we want to compute and size of Codebook size.

- Second, in this workflow, the query images are stored in “0_ImgDatasets/DatasetName/Class/Image” folder path. Then we run OnlineScript by entering the parameters, such as dataset name, detectors, descriptors, size of Codebook, and nearest neighbor to search. The results, ranking files, will be stored in Results folder.

Two different versions of the codes are available as of now: POEME_Code_1.0 (single inverted index file implementation) and POEME_Code_2.0 (Multiple inverted file implementation).