EQ2425 – Analysis and Search of Visual Data

Project #2: Visual Search System

due Sept 29, 2024, 11:59 pm

Please upload your project in a .zip file to Canvas, the .zip file should contain your project report and codes separately.

1 Objective

In this project, you will build a visual search system and evaluate it. Use SIFT descriptors as image features and use vocabulary trees as the database structures. Using query images, the goal is to recognize buildings. That is, the visual search system retrieves the objects that are most similar to the query object as represented by the query image. TF-IDF (term frequency inverse document frequency) score is used for object retrieval.

2 Image Feature Extraction

Inside the *data2* zip file, there are two folders. The images in the *server* folder are the database images. Each building object appears in three images. There are 50 building objects in total. You need to build a vocabulary tree with scoring information using the database images. The images in the *client* folder are used as query images. Each building object appears in one query image. The 50 images are taken from the same building objects in the database.

(a) Extract a few thousands SIFT features from each database image. Combine the features of the same object and save them. You may need to index your features according to the object number for later use. Report the average number of features you have extracted per object.

(b) Extract a few thousands SIFT features from each query image and save them separately. Report the average number of features you have extract per object.

3 Vocabulary Tree Construction

The vocabulary tree is built by using the hierarchical k-means algorithm. The structure of the tree is controlled by the tree branch number b, and the tree depth. Necessary information should be stored in the tree nodes for later query use.

(a) In order to query by SIFT features, what information should be stored in each node in the vocabulary tree?

(c) Based on the TF-IDF score, decide what additional information you need to store in the leaf nodes of the tree (Hint: The leaf nodes can be seen as *visual vocabularies*).

(c)) Submit the function you used to generate the vocabulary tree. Name the function as hi_kmeans(*data*, *b*, *depth*), where *data* holds the SIFT features from the database objects, *b* is the branch number of the vocabulary tree for each level and *depth* is the number of levels of your vocabulary tree (You can use the built-in k-means function in your chosen programming language).

4 Querying

Send all the descriptors of each query object into the vocabulary tree and rank the database objects according to their TF-IDF scores. Test all the 50 query objects and calculate the average recall rate. Note that our recall rate is object-based. Hence, the recall for a single object is either 0 or 1.

(a) Build three vocabulary trees by varying the settings as: $b = 4, depth = 3$; $b = 4, depth = 5$ and $b = 5, depth = 7$. For these three trees, report the average top-1 and top-5 recall rates over 50 objects.

(b) Now, use your vocabulary tree with $b = 5, depth = 7$. For querying, use only 90, 70 or 50 percent of the number of your query features. Report the average top-1 and top-5 recall rates respectively.

(c) Explain quantitatively how hierarchical clustering helps to increase the search speed when considering the number of Euclidean distance calculations.

Bonus

Propose and describe methods that can improve your recall rates. Report the comparison of the recall rates with and without your proposed method.

References

[1] B. Girod, V. Chandrasekhar, D. M. Chen, N. M. Cheung, R. Grzeszczuk, Y. Reznik, G. Takacs, S. S. Tsai and R. Vedantham, "Mobile Visual Search", IEEE Signal Processing Magazine, vol. 28, no. 4, pp. 61-76, July 2011.

[2] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), New York, June 2006.