CSE 515 (Fall2016) Phase 3 Report: Group 17

Anshuman bora, Arizona State University

vraj delhivala, Arizona State University

satyam jaiswal, Arizona State University

saumya parikh, Arizona State University

Tanmay patil, Arizona State University

**Abstract**

Finding similarity of frames between multiple videos is not enough. They also have to be listed according to some kind of importance or relevance. In phase2, we listed them by sorting with respect to the similarity index. The frame with highest similarity came first and the top *k* frames were selected as most similar sub-sequence. In phase3, we have improved the listing logic by implementing intelligent algorithms which list similar frames by significance, relevance and also by nearest neighbor importance in high dimensional spaces. SIFT vectors in a reduced space have been used as input considering they describe important visual features. Algorithms like PageRank and ASCOS++ have been used for ranking according to significance. Relevance feedback based algorithms like personalized PageRank and personalized ASCOS ++ have been used and nearest neighbor based technique of Locality Sensitive Hashing has also been used. This report talks about all these implementations in detail.

**Keywords**

**Feature Extraction, Similarity Graph, Significant Frame, Relevant Frame, Page Rank, ASCOSS, Personalized PageRank, Locality Sensitive Hashing, Multi-dimensional Indexing**

# Introduction

Terminology

TODO.

Goal Description

Phase 3 involves around experimenting with clustering, indexing, classification and relevance feedback. The detailed goal description are as follows:

1. Task1: In task1 we had to implement Video Feature Extraction algorithm where Principal Component Analysis (PCA) had to be applied to the original space to get a reduced vector space.
   1. Input: The input to this algorithm is ‘*d*’ (the target reduced dimensionality) and *‘filename.sift’* (file containing SIFT features)
   2. Output: the output would be a file named ‘filename\_d.spc’ which would contain entries in the form of *{<i, j, l, x, y>, [dim-1 , . . , dim-d]}*, such that i = video, j = frame and l = cell indices. x and y and provide the position of the SIFT keypoint and [dim-1, . . , dim-d] are the reduced dimensions. The program is also expected to output the d dimensions in the form of *<original\_index, score>* in non-increasing order of scores.
2. Task2: In task2 we have to create a similarity graph using the reduced SIFT vector space.
   1. Input: Integer *K* and SIFT keypoints in file *‘filename\_d.spc’.*
   2. Output: Graph(V, E) where V are the nodes corresponding to each frame and E are the edge pairs corresponding to <va, vb> where vb is one of the most similar frames to vb, from a different video that vb. This graph will be stored in *‘filename\_d\_k.gspc’* in the format of *<va, vb, sim(a,b)>*
3. Task3: The goal of task3 is to list most similar frames according to significance. This is to be done using 2 methods namely, PageRank and ASCOS++.
4. Task4: This task is similar to the previous task. The only difference is that the similar frames are to be listed according to relevance. The algorithms to be used are Personalized PageRank and Personalized ASCOS++.
   1. Input: Filename *‘filename\_d\_k.gspc’* and integer *‘m’* where m is number of the most significant frames to be listed.
   2. Output: Visualize the *m* frames for PageRank and ASCOS++ both.
5. Task5: This task involves implementing multi-dimensional index structures and nearest neighbor search using Locality Sensitive Hashing (LSH) tool.
   1. Input: Filename *‘filename\_d.spc’* , number of layers *L*,the number of buckets *2^k.*
   2. Output: File *‘filename\_d.lsh’* containing entries in the form {layer\_num, bucket\_num, <i;j;x;y>}.
6. Task6: A similarity-based video object search tool is to be implemented.
   1. Input: LSH index file, integer n, object in the form *{i ,j, <x1, y1>, <x2, y2>}* such that <x1, y1> & <x2, y2> is a rectangle containing the object.
   2. Output: Visualize the top n frames. Also output should contain:
      1. Number of unique SIFT vectors considered.
      2. Number of overall SIFT vectors considered.
      3. Number of bytes of data from the index accessed to process the query.

Assumptions

For this phase, we have made the following general and specific assumptions:

* Overall: Nice
* Task1: Nice
* Task3: Nice
* Task4: Nice
* Task5: Nice
* Task6: Nice

# implementation

Task 1: Video Feature Extraction

Nice.

Task 2: Video Frame Similarity Graph Generation

Nice.

Task 3: Most Significant Frame Selection

2.3.1 PageRank:

2.3.2 Ascos++:

ASCOS, Asymmetric Network Structure Context Similarity, states that the similarity score from a node to a node is dependent on the similarity score from node ’s in-neighbors to node . ASCOS++ is a similarity measure which enriches ASCOS by including the weighted paths between all the nodes of a network. Its general formula is as follows:

Where is the weight of edge e (), and , c is the discounted parameter which is used to control the relative importance of direct and the indirect neighbors. We take its value as 0.9 for our calculations. gives the similarity measure between two nodes and .

Initially we used the naïve approach for implementation. But the program was taking too much time to execute, more than an hour to be precise, to run a single iteration. Hence, we decided to move on to the distributed approach. The distributed approach uses the following formula:

Where **A=** [] is the adjacency matrix of a graph G, **P** is the column normalized matrix of **A** and **Q** = []=[]**.** This turns ASCOS++ into a classic systems of linear algebra equations, in which is a coefficient matrix with dimension n by n; is an unknown column vector with n variables to be solved; and is a constant column vector of size n.

We then apply the Jacobi iterative method to solve this system of linear equation. We use the Jacobi because this approach possesses high degree of natural parallelism for distributed computation.

Task 4: Most Relevant Frame Selection

Nice.

Task 5: Multi-dimensional Index Structures and Nearest Neighbor Search

Nice.

Task 6: Similar Video Object Search

Nice.

# interface specification

Nice.

# system requirements

Nice.

# related works

PageRank

PageRank algorithm is heavily used in a wide range of fields. It is used in recommendation systems like Amazon to recommend shoppers about their next purchase and Netflix users about which movie to watch next. Another major use of this algorithm is in ranking of tweets in the social networking site Twitter. [51]

ASCOS++

ASCOS++ is a measure which has very limited application so far. As this measure captures the similarity scores among any pairs of nodes in a network, it has been used in discovering similar objects in a social network. [521]

Locality Sensitive Hashing

LSH is used in Near Duplicate Detection, an approach in which is used in web crawling to determine similar kinds of websites. [531] It is also used for hierarchical clustering of data, an approach which seeks to build a hierarchy of clusters, either bottom up or top down. [532] LSH is also heavily used in Acoustic fingerprinting, a process in which audio signals are converted into a condensed state which helps in finding similar audio samples from a database.[533]

# conclusion

Nice.

**BIBLIOGRAPHY**

51. Goel, Ashish. "Applications of PageRank to Recommendation Systems." *Stanford*, web.stanford.edu/class/msande233/handouts/lecture8.pdf. Accessed 30 November 2016.

521. Hung-Hsuan Chen , C. Lee Giles, ASCOS: an asymmetric network structure COntext similarity measure, Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, August 25-28, 2013, Niagara, Ontario, Canada [doi>10.1145/2492517.2492539]

531. Gurmeet Singh Manku , Arvind Jain , Anish Das Sarma, Detecting near-duplicates for web crawling, Proceedings of the 16th international conference on World Wide Web, May 08-12, 2007, Banff, Alberta, Canada [doi>10.1145/1242572.1242592]

532. Koga, Hisashi, Tetsuo Ishibashi, and Toshinori Watanabe (2007), "Fast agglomerative hierarchical clustering algorithm using Locality-Sensitive Hashing", Knowledge and Information Systems, 12 (1): 25–53, doi:10.1007/s10115-006-0027-5 .

533. “dejavu - Audio fingerprinting and recognition in Python”. Github, https://github.com/worldveil/dejavu/. Accessed 30 November 2016.