CSE 515 Multimedia and Web Databases

Phase #3

(Due December 1st 2016, midnight)

Description: In this project, you will experiment with

- clustering,
- indexing,
- classification, and
- relevance feedback.

This project will build on the deliverables of the previous phase. In particular, let us assume that $filename.sift$ file has been already created for a given set of video files and parameter settings.

- **Task 1: Video Feature Extraction:** Implement a program which takes as input
  - a target dimensionality d and
  - the descriptors, with their orientations and scales, of the keypoints in the file, $file\_name.sift$,

  and applies PCA to map these SIFT keypoints into a d-dimensional vector space.

  The results are stored in an output database file , $filename_$d.spc, with entries of the form

  $$\big\{\langle i, j, l, x, y\rangle, [dim_i, \ldots, dim_d]\big\},$$

  Here i, j, and l are the video file, frame, and cell indices, respectively; the pair x and y together provides the position of the SIFT keypoint in the frame.

  The program also reports the d dimensions in terms of the input vector space. The d dimensions are to be reported in the form of $\langle original\_index, score\rangle$ in non-increasing order of scores. You can use a Matlab package for PCA.

- **Task 2: Video Frame Similarity Graph Generation:** Implement a program which takes as input
  - an integer k and
  - and the SIFT keypoints stored in the file $filename_$d.spc

  and creates a similarity graph, $G(V, E)$, where
  - V corresponds to the frames in the videos in the directory and
  - E contains edge pairs $\langle v_a, v_b\rangle$ such that, for each video frame v_a, v_b is one of the k most similar frames to v_a that is not already in the same video file.

  Results are to be stored in a file, $filename_$d_k.$gspc$, where each line is of the form

  $$\big\{v_a, v_b, sim(a, b)\big\},$$

  where $v_a = \langle i_a, j_a\rangle$ and $v_b = \langle i_b, j_b\rangle$ are two frames and $sim(a, b)$ is the degree of similarity between these two frames.

- **Task 3: Most Significant Frame Selection:** Implement a program which takes as input
  - the graph file $filename\_d\_k.gspc$ and
  - an integer m,

  and identifies the most significant m frames in the collection using (a) PageRank and (b) ASCOD measures. See

  > Sergey Brin , Lawrence Page, The anatomy of a large-scale hypertextual Web search engine, Computer Networks and ISDN Systems, v.30 n.1-7, p.107-117, April 1, 1998

  and

  > Hung-Hsuan Chen and C. Lee Giles. 2015. ASCOS++: An Asymmetric Similarity Measure for Weighted Networks to Address the Problem of SimRank. ACM Trans. Knowl. Discov. Data 10, 2, Article 15 (October 2015)

  Visualize the selected m frames for both approaches.

- **Task 4: Most Relevant Frame Selection:** Implement a program which takes as input
  - the graph file $filename\_d\_k.gspc$,
  - an integer m, and
  - three input frames (each described as a pair of video/frame ids)

  and identifies the most significant m frames (relative to the input frames) using
  - personalized PageRank measure

    > Huang, S., Li, X., Candan, K. S., Sapino, M. L. (2016). Reducing seed noise in personalized PageRank. Social Network Analysis and Mining, 6(1), 1-25. [6]
  - and a suitably modified version of ASCOS to account for seed frames.

- **Task 5: Multi-dimensional Index Structures and Nearest Neighbor Search:** Implement a Locality Sensitive Hashing (LSH) tool, which takes as input
  - a file $filename\_d.spc$,
  - the number of layers, L, and
  - the number (2^K) of buckets per layer,

  and maps each keypoint into a hash bucket for each layer. See

  > Alexandr Andoni and Piotr Indyk. "Near-Optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions". Communications of the ACM, vol. 51, no. 1, 2008, pp. 117-122.

  The tool should output the results into a file $filename\_d.lsh$ containing entries of the form

  $$\{layer\_num, bucket\_num, \langle i; j; l; x; y\rangle\}.$$

- **Task 6: Similar Video Object Search:** Implement a similarity-based video object search tool which takes as input
  - an LSH index file, $filename\_d.lsh$,
  - an integer n,

– an object described as

$$\langle i; j; \langle x_1, y_1 \rangle; \langle x_2, y_2 \rangle \rangle,$$

where i and j are the video and frame numbers respectively and $\langle x_1, y_1 \rangle; \langle x_2, y_2 \rangle$ is a rectangle containing the object,

and outputs (and visualizes) n frames (that are not in the same video of the query) that contain the most similar video objects.

The program also outputs (a) the number of unique SIFT vectors considered, (b) the overall number of SIFT vectors considered, and (c) the number of bytes of data from the index accessed to process the query.

Deliverables:

- Your code (properly commented) and a README file.
- Your outputs for the provided sample inputs.
- A short report describing your work and the results.

Please place your code in a directory titled “Code”, the outputs to a directory called “Outputs”, and your report in a directory called “Report”; zip or tar all off them together and submit it through the digital dropbox.