

# Content Based Video Retrieval (CBVR)

---

*Course Project: CS452 / EE 512 Digital Image Processing*

## *Task 1: Baseline Algorithm for CBVR*

### *An extension of Content Based Image Retrieval (CBIR) for CBVR*

In this task you are required to extend your existing algorithm of CBIR that you developed in your first assignment for CBVR. There are two baseline algorithms that you have to develop. Video-in-video-out and Image-in-video-out

## **1 Video-in-video-out**

### **1.1 Database creation**

In the CBIR assignment you created a database containing a feature representation of each image. Each image was segmented into blocks and then histograms were computed for each of those blocks. The concatenation of those histograms was then used as a feature vector representing that particular image. Video can be considered as sequence of images with temporal consistency between them. In this task you are not required to exploit the temporal consistency instead each image of the video will be treated as an independent frame.

You are required to compute the same feature vector as performed in the previous assignment of CBIR. However, instead of computing the feature vector for single image, you are required to compute a feature vector for multiple images within each video. Only process first 3 minutes of each video i.e. total 180 seconds of video content. You should compute your feature vector for an image after every 15 seconds i.e. for total of 12 images per video ( $3 \times 60 / 15$ ). This would result in a feature vector of length  $9 \times 3 \times 256 \times 12 = 82944$  per video.

You can further reduce the length of feature vector by using techniques discussed in previous assignment on CBIR. Note this feature vector may not be very robust, here one of the purpose is to get your hands on with processing of video content.

Since the database consists of wide variety of videos in various different formats which Matlab may not be able to handle. To further simplify things you are only required to process AVI files (i.e. files with .avi extension) and can ignore videos in other formats like \*.mkv, \*.mp4 etc. You can work on any 3 or more shows of your choice or can get the data from us.

### **1.2 Video Retrieval**

The base-line video retrieval algorithm will remain the same as used in CBIR. Instead of comparing a feature vector of length  $9 \times 3 \times 256 = 6912$  or its quantized versions, here you will be comparing feature

vectors of length  $6192 \times 12 = 82944$ . Depending upon the quality of your code in assignment on CBIR, you may not need to change anything in the retrieval part except that instead of showing images as results you will be displaying representative image from the video along with the file name on a figure window.

## 2 Image-in-video-out

### 2.1 Database creation

This task is slightly more difficult than the previous naïve approach. In this task you are first required to divide the video into shots. Then select say 3 representative images from each shot and create a feature vector for each image of the shot as done in your CBIR assignment.

### 2.2 Video Retrieval

The base-line video retrieval algorithm will remain the same as used in CBIR exactly the same as in case of CBIR. The difference will be you will be comparing feature vector of a single input image with the representative image of each shot in the video. This will be a one-to-many mapping resulting in multiple matching scores for each video. The final result could be based on either the cumulative score (such as mean) or just by max score.

## 3 Survey of Related Work

Additional to the baseline algorithm for CBVR, an important task at this stage is to get the understanding of existing literature on CBVR. I have selected three papers that cover some aspect of the CBVR space which are listed below and also uploaded on LMS.

1. A Survey on Visual Content-Based Video Indexing and Retrieval , IEEE Cybernetics, 2011  
<http://www.dcs.bbk.ac.uk/~sjmaybank/survey%20video%20indexing.pdf>
2. Large-Scale Near-Duplicate Web Video Retrieval: Challenges and Approaches, IEEE Transaction on Multimedia, 2013  
<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6530586>
3. Tiny Videos A Large Data Set for non parametric Video Retrieval and Frame Classification, IEEE Transaction on Pattern Analysis and Machine Intelligence, 2011  
<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5487525&isnumber=5692151>

You are also free to perform your own search on Internet, Google Scholar, IEEE Explorer etc. If you come across a good paper feel free to share among other classmates.

You are also required to submit a short report discuss your survey on CBVR and an overview of your approach for Task 2 on near-duplicate detection and using a spatio-temporal feature within a spatio-

temporal cuboid. You must provide a flow diagram of your approach for Task 2. There is no limit to the length of the document, shorter the better (upto 4 pages could be enough with normal formatting). Additional marks will be given on a graphical or tabular representation of existing approaches.

## 4 Submission

- Submit a folder containing all your m-files, database files and a “runme.m” file that test each of your implementation
- Submit a document explaining your methodology i.e. how you computed the feature vector and what is the rationale behind it. Also provide a discussion on analysis of results.
- Submit a separate document on Survey of Related Work
- Submit a folder containing snapshot of all your results
- Folder containing additional references if any
- Please also check late submission policy on LMS

## 5 Appendix

- In case you have issues in reading the videos in Matlab you can convert video format using Virtual Dub software (<http://www.virtualdub.org/> )
- To open a video in Matlab you can use `mmreader` and `read` functions. The code below extracts 12 frames from first 3 minutes of a video.

```
clc; clear; close all

filename = 'K:\Datasets\Heroes [1x22] Landslide.avi';
readerobj = mmreader(filename);
fps = get(readerobj, 'FrameRate');
frno = int32(linspace(fps, fps*60*3, 12));
for i=1:numel(frno)
    myframe = read(readerobj, frno(i));
    imshow(myframe);
    pause
end
```