Video Browsing and Indexing

CSCI 576 Project Spring 2012

Instructor Parag Havaldar

With the advances in inexpensive digital video (and audio) capture devices, media data is getting to be common place now. There is a lot of digital video information everywhere – streamed to you via the internet and cable networks, hosted on websites, your own personal hard disks etc. With a lot of video information, there has to be a way to browse through it efficiently, and search through it effectively. This is already available in many ways for simple media types such as text and images. For instance, if you load a big text document, and want to quickly read about information on a specific topic, you can easily search through the entire text using a "search string", and you can just as easily bring up other documents that contain the same text string. For images, there have been numerous metaphors to view collections of many images, but searching based on an image query is only a recent technology that, unlike searching/indexing text, has yet to gain reliability under all circumstances. For an even more complex media type such as video, this is not straight forward at all.

The motivating question here is two fold, how you can develop a system that should:

1. *quickly browse* a video in order to extract the most useful information that the video has to offer (easier!)
2. *efficiently* index and search for contextually similar videos or video segments in other videos (harder!)

A few example scenarios where video browsing/indexing is desirous are:

- You watch a video recording of a sports game, which is two hours long, but you want to do this in a few minutes skipping only to the interesting sections. Furthermore, if you see a game play that is interesting or entertaining, you might want to see similar game plays in other matches.
- You want to quickly browse a surveillance video to see interesting events that might have happened. If you find an interesting event, you want see where else that event has occurred in the same or other recorded surveillance videos.
- You want to watch a movie ninety minute feature in ten minutes. Furthermore, if you like some action in that movie, you might want to see other movies with similar actions.
- You want to go through a documentary or an video interview of a person but only for topics that are relevant to you. And if you do find some interesting video footage about a topic, you want to quickly find other videos (or video parts) that discuss the same or similar topics.

Clearly the problem is very useful, undoubtedly practical but also fairly complex and perhaps really hard to get working correctly for all different kinds of videos. This is because of many reasons – one user's interpretation of context in a video or video segment might not be another user's interpretation, ultimately yielding to different search pathways and results. Understanding "context" can be very difficult since it involves quantitative analysis such as color, motion, objects, sound levels and also qualitative analysis such as semantic understanding, pragmatic interpretation. etc. Consequently no results of a search are wrong, but some may be contextually more correct. In order to make it a worthwhile semester project, let's reduce the problem space by restricting the definition of context and work with a very simplified set of videos. We will restrict our search methods to use only media metrics that we have learnt in class such as color analysis, motion statistics, audio statistics etc. This will help generate results that are can be compared in a measured manner.

There are three parts to the project: You will be given many videos in the .rgb format and corresponding audios in the .wav format. Please see DEN class website for exact particulars (image size, frame rate, audio sampling rate etc.) You can assume that these media properties will remain constant for all media elements given to you. Here is a list of concrete tasks that your video browsing/indexing project needs to achieve, each is explained in more detail below:

1. You are to write a simple A/V interface to load a video/audio file – play it in a synchronized manner. You will need to show a pre-computed image map that allows for easy interactive browsing for the audio/video media.
2. Offline processing - (not real time), you need to develop a set of metrics that scan the video/audio and extract quantitative descriptors based on concepts learned in this course – you are welcome to add more if you feel. These descriptors will be used to index/search into other video segments.
3. Extend your A/V interface to search/index and show relevant indexed results.

A/V Player and Interface

Design a simple GUI and interface to play videos, index into video and search similar video segments. You will invoke your player by invoking

MyProject.exe myVideo.rgb myAudio.wav

The GUI below is a simple interface, and is only a suggestion, you are free to develop any form of GUI – as long as all the functionality is present –

1. Play – plays the video audio in synchronized manner
2. Pause – pause the current video audio file
3. Stop – stops the current play and resets to the beginning.
4. Search – to index and find video segments with similar context/content.

Here is an example instance at play when you start:

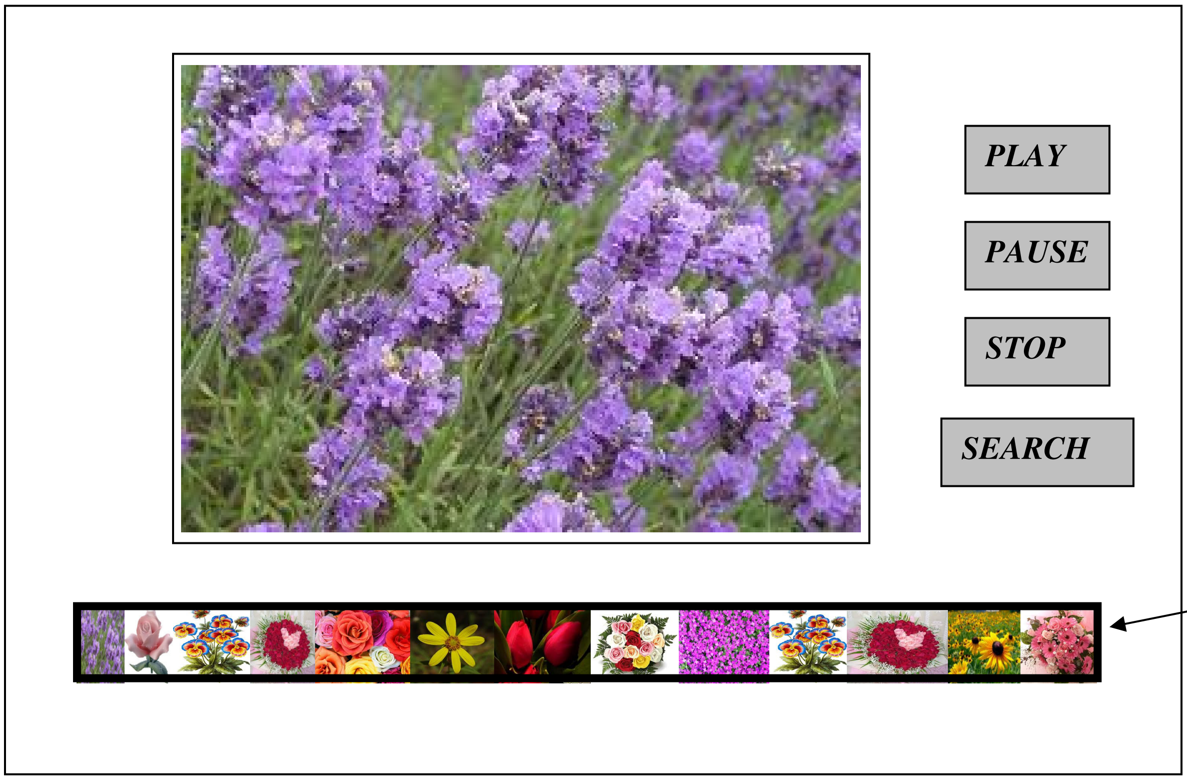
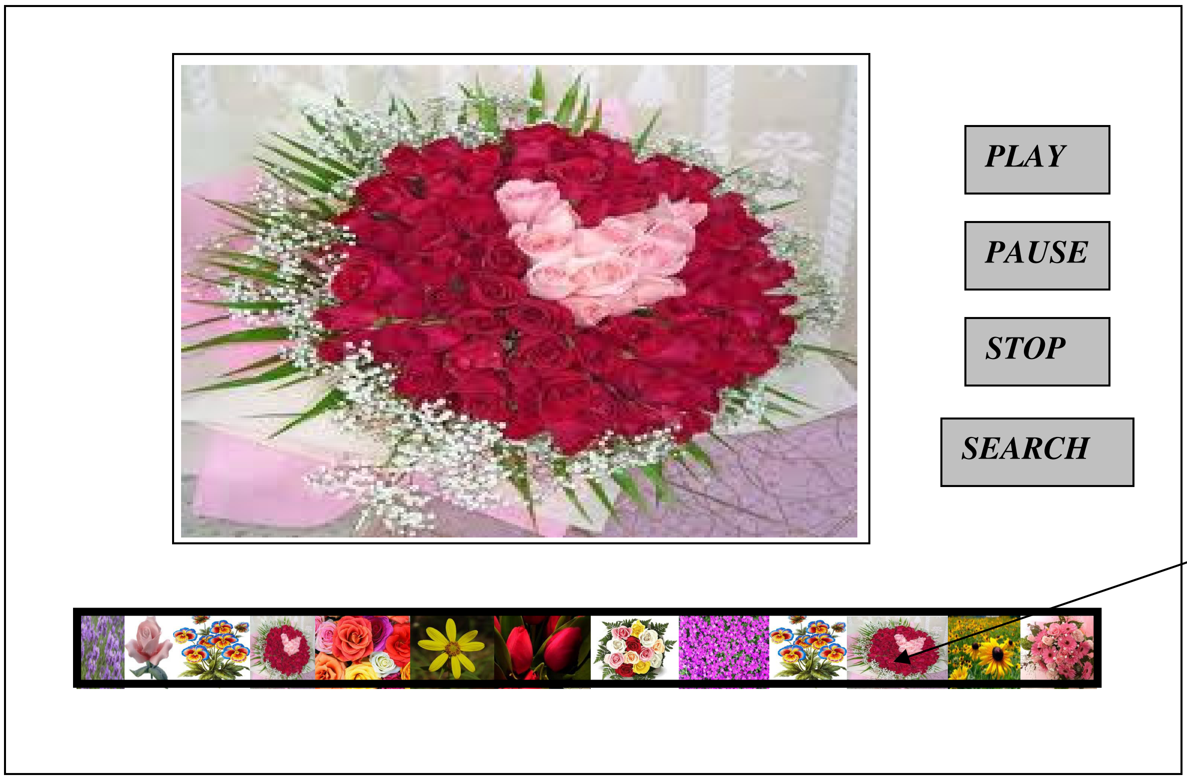


Image strip representing all of the loaded video



Clicking on the image strip here should start playing the video from the appropriate frame index

Offline processing to extract Descriptors & Image Strips:

This is an offline step that should be run prior to running your viewer. There are two outputs you need to generate per video – in a format that you desire.

1. Generating a representative image strip:

Given a video, you will need to generate a small image strip that represents some/all of the frames in the video. Examples are shown in images of the interface. This will be used in the player to visually see a representation of the video and browse into the video by interacting with the image strip. At the simplest level the image strip can be generated by a set of sampled frames, or parts of frames that are scaled down and pasted together. This will help browsing and interacting with the video.

2. Extracting descriptors:

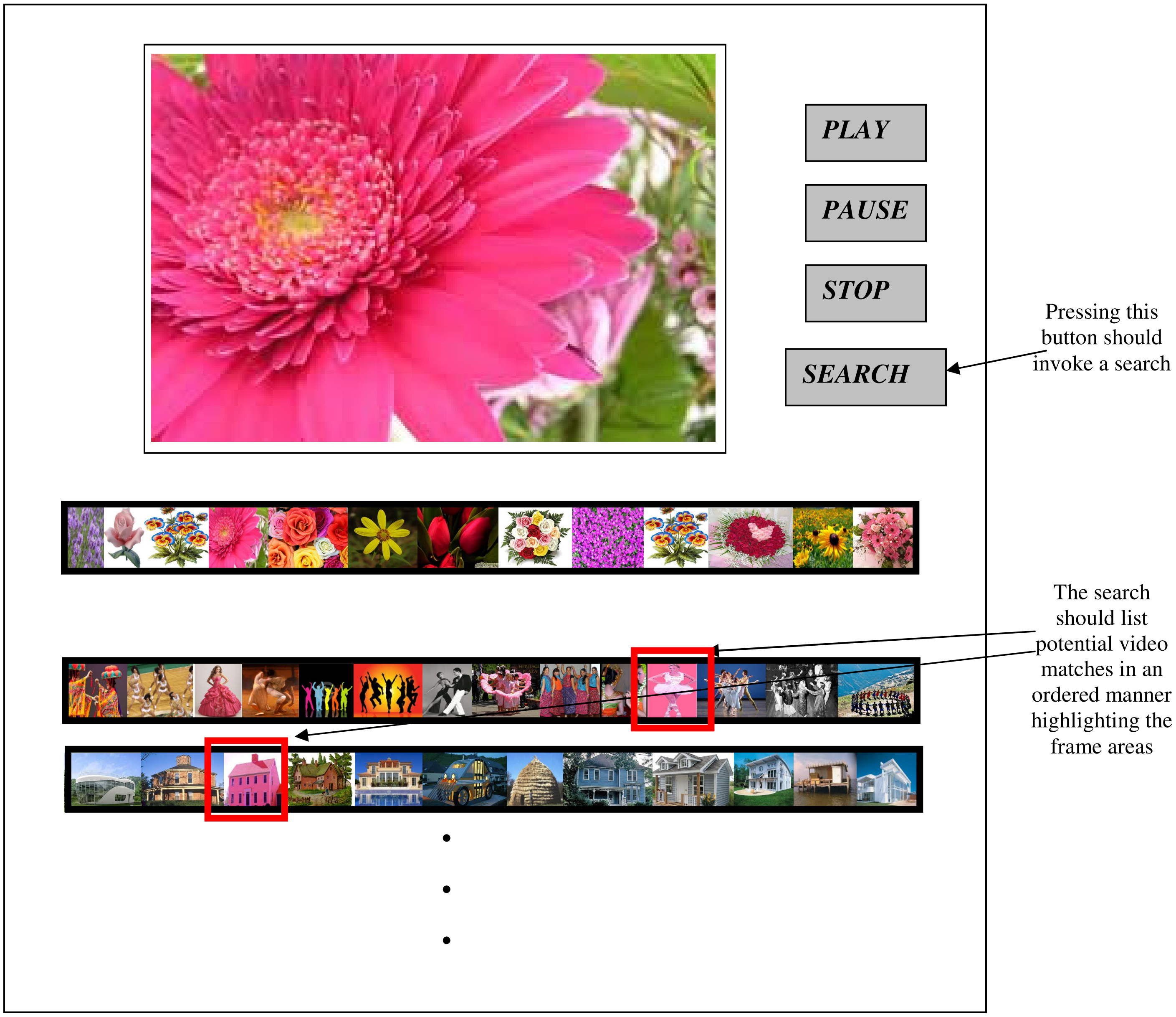
Implement a process that will go through all video/audio systematically and extract descriptors based on concepts learned in class. The descriptors are quantitative numbers that are extracted per frame and organized into a representation for searching and indexing. Suggestions are listed below, please give thought to how you would extract them, and organize them to be used for searching.

- Color – For every video you can find a color theme. Eg extracting the most dominant color per frame as a list of number and collating this information hierarchically
- Motion – For every video you can compute motion statistics. Every frame can given a quantitative number depending on how much motion you perceive in that frame.
- Sound – Based on the wave PCM samples, you could compute audio threshold levels or even frequencies that give allow you to compute a quatitative assessment of how much sound/frequency there is in an audio segment.

Extending A/V interface for indexing:

You will need to extend your A/V interface to perform relevant indexing functions based on descriptors extracted in an offline step. Given a frame of the video (and corresponding audio) that is of interest to you, you want to search for “similar content” in other videos. The search should return a list of matched videos in some weighted order. The interface should also highlight the matched frame areas as shown below in red. Clicking on the area (or any other area) should play the video from that point.

Remember - the matches may not be logically correct, but should be reasonably accurate based on the search contexts used.


PLAY
PAUSE
STOP
SEARCH
Pressing this button should invoke a search
The search should list potential video matches in an ordered manner highlighting the frame areas