CSCI 576 Multimedia Project

Instructor: Parag Havaldar

Demo date: Wednesday Dec 6th & Dec 7th 2017

The course project is meant to give you an in depth understanding of some of the areas in multimedia technology. Since this is a broad field, there can be a variety interesting projects that can be done depending on your interests which can also extend to related and complementary topics that are taught in class.

Also, I have often found that a large project can be successfully accomplished via collaboration. Additionally, working together to design and integrate code can be a rewarding exercise and you will frequently need to work in teams when you set out to work in the industry. Accordingly, please form groups of **exactly two** students. We have started a discussion board to help you make groups, where you may post your preferred language of implementation, availability etc.

This time I want to suggest a topic that is at the heart of trying to quickly browse, interact with and understand AV content. These days there is no dearth of prolific video content that people create for themselves (video of family, friends, personal events etc.) or available commercially such as movies, documentaries, sports videos as well as hobby and educational videos available on social media sites. The metaphors of linear browsing, that is watching the video as we are traditionally used to, or even trick play (fast forward) available for watching DVDs might not suffice given that there lots of content to watch. You need a way to quickly visualize what the video content is about and then make a more informed decision to explore it, quickly see relevant details before you decide to invest your time into watching it. The question then is what kinds of viewing or exploring metaphors can you come up with that allow this. This is a topic of much research and while there are no correct or incorrect answers, surely some might work better than others.

Creating and browsing videos using “summary” images

In this project, you will implement a *video summarization* algorithm that produces a *“summary image”* summarizing the video content. Furthermore, given a video and an audio file, you are required to design a user interface that can

- display the video/audio stream in synchronization and
- given the “summary image”, allows a user to interact with this summary image to appropriately jump to the correct location in the video in order provide an effective visual browsing interface.

With this interface, users should be able to quickly browse the video/audio content, and conveniently hone down on areas that they are interested in, quickly explore and hence efficiently determine whether the content is worth further viewing or watch only the relevant parts that interest them.

INPUTS AND OUTPUTS

Input to your Process:

1. A video file in CIF format (352x288) (*.rgb)
2. A audio file in WAV format, synced to audio

More details regarding frame rate and length of video will be given in data description file. You may assume these parameters will be the same for all files used in this project. Sample input data can be found here:
https://drive.google.com/drive/folders/0BxQdeWBJBOg1dkZvRTN3R3B0SFk

Expected Output:

1. A summary image (or a hierarchy of images) for the given video. This can be an offline process
2. You are also required to design and implement an interface that loads the video/audio and summary image and allows to explore the visual content. You should be able to play video and audio synchronized as you explore the video with your summary image. Step 1 should also create appropriate pointers/data structures to help the interface index into browsing the A/V content

Example Invocations:
CreateSummaryImage.exe MyVideo.rgb MyAudio.wav – generates *MySummary.rgb*
ExploreVideo.exe MyVideo.rgb MyAudio.wav MySummary.rgb

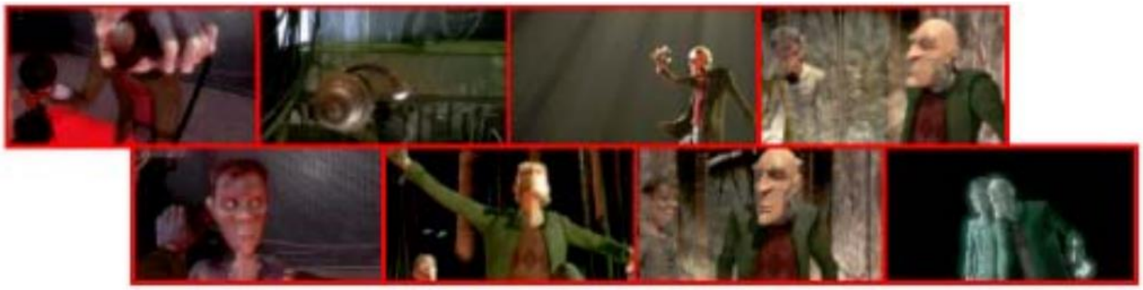
IMPLEMENTATION STEPS

1.Video/Audio Synchronization
You will be given two separate files: one video and one audio file. You are required to synchronize the two files as a movie.

2. Key frame extraction and use of key frames to create a summary image

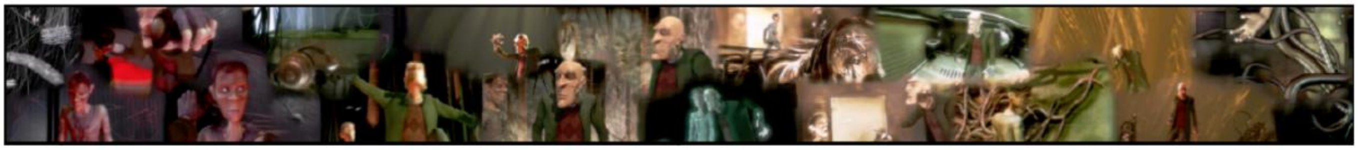
The easiest (but not so descriptive) way to extract key frames is to select one frame from each *n* frame (e.g., *n* = 100 then select the first of every 100 frames). This might be good enough to get a first start at understanding the working of this project but for a summarization purpose, this might not be a good algorithm because every *n*th frame may not correctly depict interesting frames in the video.

There are better ways to describe key frames instead of every *n*th frame. One way to do this is to sample one frame in each scene change, or cuts because all cuts are logically continuous. A more statistical representation may be to choose key frames where the motion information content in the video might be high – such as objects are in motion, or the audio levels are high. This will need you to analyze video frames using motion prediction taught in class, or analyze samples of audio for sound levels. You are free to use any frame extraction techniques and smarter techniques will generally lead to a more descriptive summary image Please see our evaluation policies below. An example result of this step might take a form as shown below

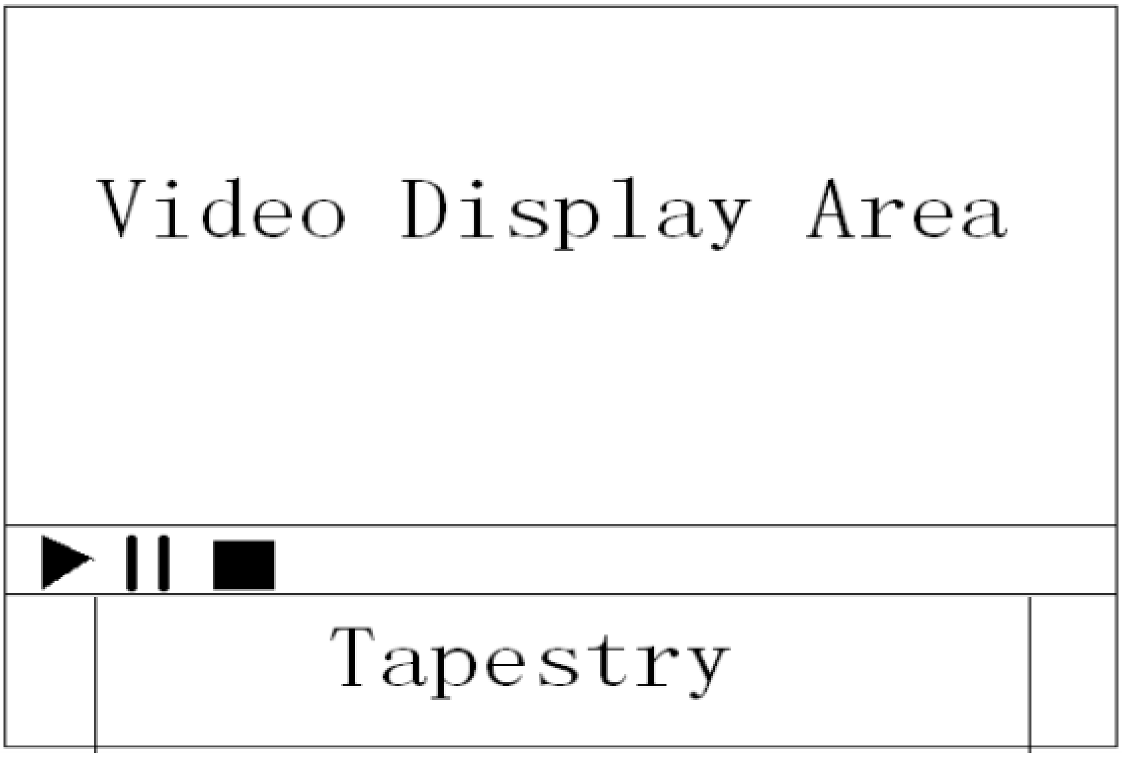


3. Key frame blend
After extracting all the key frames, we have a block-like sequence of frames. You could create a summary image by concatenating these frames together side by side, but that might make a long uninteresting image. A better visual could be obtained by eliminating boundaries between frames by selecting interesting sub areas in all frames and putting them together. For this process, you could use a variety of techniques for this (eg context based sub sampling, seam carving etc.). You can also treat this problem as an optimization problem, please refer to [2] [3] for more details. Again, please see our evaluation policies below to determine which techniques to use.

OUTPUT
1. A binary file representing the summary image in RGB format that summarizes the video content. The summary image is n (say 100) pixels in height, and m pixels in length (an appropriate *m*, which depends on how many key frames or detail of content you do decide to express) (A complex example of such an images is shown below [1])



2. Pointers that continuously map a location on the summary image to a frame location on the input stream – the object (or area) being that clicking at a location on this summary image should start playing video/audio from that location onwards. You are trying to use the generated summary image as a “visual” guide to understanding and exploring what the video is about. Note that the structure you choose to represent this is to be defined by you – such as an additional metadata file, or you may choose to embed the pointer data in the same file. Specifically, once the user selects a place in the summary image, the program should be able to quickly index into the original video and start playing the video from that time onwards with the audio correspondingly synchronized.
3. A user interface that synchronizes the video/audio files and plays the video. It should at least contain three buttons - *play*, *pause* and *stop*. The summary image is display at the bottom of the interface for users to browse and explore the video. Once the users click on a position of the summary image, your interface should locate the place in the original video, and start to play from there. A simple example of the interface is shown below:



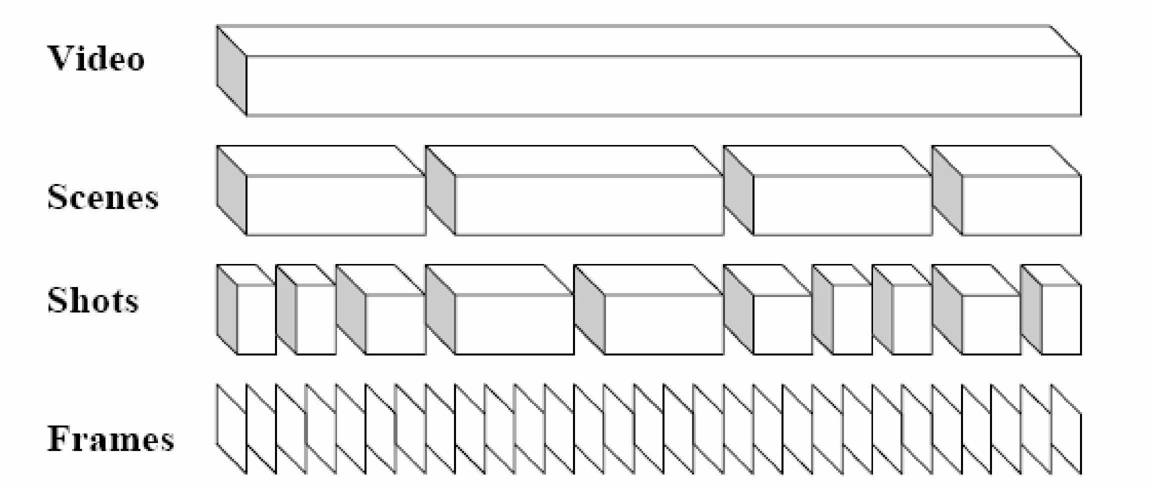
PROJECT EVALUAITON

When your program is launched, it should load the audio/video file and the summary image along with the metadata pointers. An example of such an interface in shown above NOTE: the computation of your summary image will need a lot of time and so you may do this as an offline process. But once you generate the summary image and the metadata, you should be able to "interact" with it in real time.

Your project will be evaluated on
1. The real-time of your audio/video synchronization
2. The correctness of how you handle the interactivity – whenever you click on a position in the summary image, it should start playing the video at that location.
3. The quality, completeness and continuity of your summary image
4. The ability to answer questions related to your implementation and the theory around it.
Additionally, for EXTRA CREDIT -
5. Create a multi-resolution summary image so that if the user so desires you can shift between a level 0 images (having less detail) to a level *n* image having more detail, and continue having a finer level of browsing

REFERENCES
[1] http://www.cs.princeton.edu/gfx/pubs/Barnes_2010_VTW/index.php
[2] http://www.wisdom.weizmann.ac.il/~vision/VisualSummary.html
[3] http://www.cs.princeton.edu/gfx/pubs/Barnes_2009_PAR/index.php

Anatomy of a video:



- **Frame**: a single still image from a video, eg NTSC - 30 frames/second, film – 24 frames/second
- **Shot**: sequence of frames recorded in a single camera operation
- **Sequence or Scenes**: collection of shots forming a semantic unit which conceptually may be shot at a single time and place