Fall 2023 CSCI 576 Multimedia Project

Instructor: Parag Havaldar

Demo date: Wed December 6th, Thu 7th, Fri 8th 2023

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 of 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, larger projects can be successfully accomplished via collaboration with multiple students working in a group. Additionally, working together to design and integrate code can be a rewarding exercise and you will frequently need to work in teams when you graduate and work in the industry. *Accordingly, please form groups of **at least three**, but **utmost four** students.* If you need help forming groups, please use Piazza discussions, where you may post your preferred language of implementation, hours of availability etc. Once your group is decided, please send the TAs an email so we with the organization around the project and project grading. Remote DEN students may form groups with in-class students. If you are having trouble finding a partner, please send an email to the TAs and we will try to facilitate group formation. *The demonstrations will be done online over zoom where all members must be present and on camera for Q&A.* You will be asked to submit code in certain cases for further evaluation.

This semester, we are proposing a relevant project to search and index into video/audio with interfaces to show your results in an interactive media player. Details are explained on the following pages. For this project, we are placing no restrictions around the language of use or libraries to use. You are welcome to use external libraries, environments of your choice – as along as you are able to fulfil the objective of the project evaluation.

Searching and Indexing Video/Audio

Simple media types like text have an excellent metaphor to search and index into. For instance, when you open a pdf/word document or book, you can easily search using a text string or a sentence. This search results in an exact position where that sentence lies in the document. Such searches and indexing mechanisms help with navigating and browsing digital text documents. In a similar manner, automated programs can take text strings/sentences as input and search into a database of documents to find relevant documents (and indexes within the document) where the input string/sentence is present. This is not so straightforward with other media types like video and audio.

There are commercial applications that can search into and "identify" music based on a short sample of audio – eg "Shazam", where a small 20 second snippet of a song, no matter if it is the intro, verse, or chorus, it will create a fingerprint for the recorded sample, consult the database of preprocessed audio music, and use its music recognition algorithm to tell you exactly which song you are listening to. There are also similar other apps for indexing music given a small 20 second audio or even written or sung lyrics – Genius, Musicxmatch, SoundHound, MusicID, Soly etc.

When it comes to similar functionality for videos (with audio), the metaphors for searching and indexing are not that easy. Most of the successful commercial applications are limited to using YouTube (or some similar websites) where, search is accomplished using an input text string which is *text* matched with metadata strings for all the video to find the match. The metadata strings are either user generated or autogenerated using closed caption or audio-to-text mechanisms. Querying a video with a video is one less explored metaphor, which is the focus of this project.

In this project you will be given a database of videos (with synchronized audio). These videos will be long and will have playtimes upwards of ten mins. The main task of your project is given a video (with synchronized audio) snippet of 20-40 seconds, find the matched video and also compute the indexed start frame of the query video into the matched video. As you may well understand, this is a hard problem to solve in the general sense. But we will make this easier by putting restrictions and narrowing the scope. You may assume:

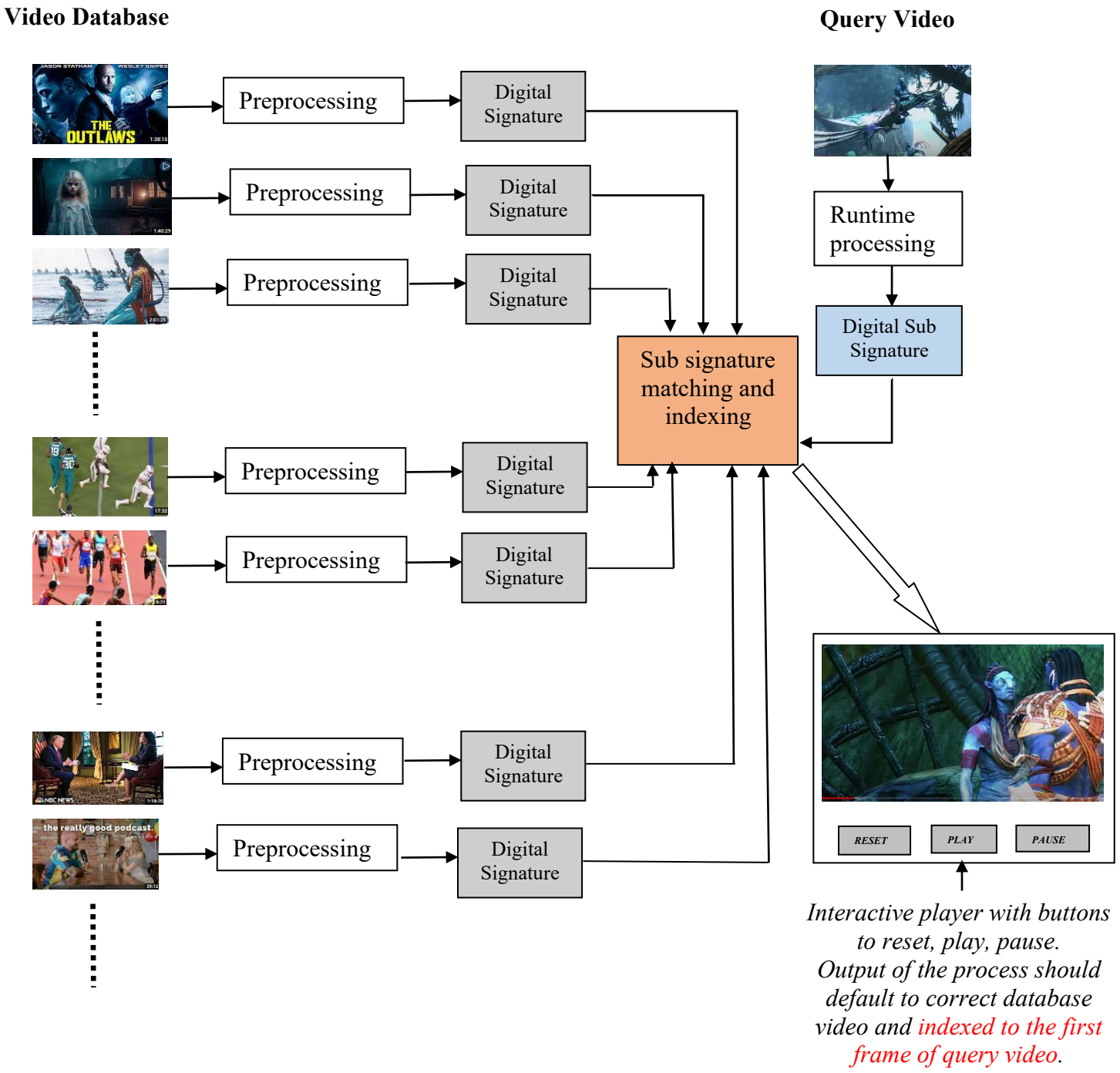
- All database videos are similar in format (width, height, fps) though exact playtimes may vary. The content of these videos will be discriminatory and different.
- The 20-40 second query video snippet used will also be of the same format as the database videos. Furthermore, the content of the query video will be an ***"exact"*** match to some content in a video in the database.

Since the query video (with audio) frame sequence is guaranteed to be an exact match in some database video (with audio), one inefficient way to solve this is a brute force comparison of a moving set of windowed frames in each database video with the window of query frames until a least error match is found. *But a media aware process can definitely perform more efficiently.* One thought might be to preprocess (in an offline manner) your videos to create a digital signature for each video. At demonstration time, you will similarly process your query video to

create a sub-signature. The problem then comes down to finding the best match of the sub signature to a signature – which is essentially a pattern matching problem.

Your project will be called as follows: "*MyProject.exe QueryVideo.rgb QueryAudio.wav*"

The figure below outlines a broad description of a possible processing pipeline.



The broad steps for your project include:

Preprocessing to compute digital signatures:
Here you are asked to go through a entire video and create a digital signature. There are many ways to create such signatures based on descriptors which should include (but not limited to):

- Shot boundary details: Shot boundaries where one shot transitions discontinuously and abruptly to a new shot is good key indicator to create matching signatures. Any video structurally consists of shots. Computing shot boundaries can help with a fast indexed search rather than a brute force search. For example, if you detect two shot boundaries with a certain number of frames in between them in your query video, then you can look for hypothesis matches in your database videos that follow the same pattern. Confirmation can then be made by comparing the video and audio data at the exact frames. You will need to address cases with different numbers of shot boundaries. If there are no shot boundaries detected in the query, then you may need to resolve to other metrics given below to compute a signature.
- Color – For every video you can find a color theme. Eg extracting the most dominant colors per frame as a list of numbers and collating this information.
- Motion – For every video you can compute motion statistics. Every frame can be 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 quantitative assessment of how much sound/frequency there is in an audio segment.

We leave it to your analysis and experimentation to understand the best way to create digital signatures. You will also need to create a digital signature for your query video, and since they will be shorter than the actual video, let's call it a digital sub-signature.

Sub signature matching:
Given all the signature, which are ultimately patterns of numbers, create a process to match the sub signature within all the signatures to arrive at the best matching one. The output of this process should point to the correct video and the exact frame offset in the video that corresponds to the first frame of video. You may one or multiple matches depending on thresholds used, which may then be evaluated for confirmation using the actual video and audio data.

Displaying the output:
The output displayed needs to be in a custom video player with basic functionality and support audio/video synchronization. It is expected to default to the first frame of the query video. Basic functionality includes play (from current frame), pause while the video is playing and reset to go to the beginning of the video. You are free to use any UI mechanism of choice, as long as you support the basic requirements.

Expectations and Evaluations:

We do expect that you analyze the problem space and produce a solution. The answers here are not subjective since there will always be an exact match. The evaluation will include:

- The correctness of your match (both the actual matched video and the indexed offset in it to point to the query video)
- Ability to show audio-video synchronization when rendering.
- The time your algorithm takes to find a correct match.
- Oral Q&A about your algorithms and process.