Report: Final Project
Multi-camera View Selection

Course:	Computer Vision
Professor:	Andrea Cavallaro
Submission:	30, Aug, 2019
Team 4:	Taeyong Song Jungin Park Wonil Song Eungbean Lee

1. Problem Statement

In this project, we are given with a blackbox system that takes multiple videos (3 or more) of a single event, with different viewpoints. The system automatically aligns the videos over time and selects the best view to describe the scene at each moment. We are to analyze the system by testing the system with various input videos and observing the outputs. We address the limitations of the system, then discuss and show how we can achieve improvements by adding/modifying the module(s).

To this end, we first explain the options that we considered for the algorithm inside the system and how we set up some scenarios to check how the system works. Then, through the observations between the inputs and outputs, we address the limitations of the system that we would like to alleviate.

1.1. Scenarios to Test the System

We first set up the options of the algorithm that system may use for temporal alignment (synchronization) of the videos and viewpoint score calculation. For the temporal alignment, three candidates of i) audio, ii) visual, or iii) audio-visual methods are available. Similar to the alignment method, we assumed that the system may use audio or visual cues for viewpoint score calculation.

To test the system, mainly with the synchronization algorithm, we shot videos of different scenarios, as illustrated in Fig. 1. In an indoor environment during nighttime, a person walks around i) carrying a smartphone with a music on, ii) as quite as possible without music, iii) sound is artificially removed from i), and iv) with music and lights are turned off. Scenario i) considers a general scenario where both audio and visual cues are sufficiently provided. Scenario ii) and iii) consider where the audio cues are insufficient or not existing. Lastly, scenario iv) assumes the case where visual cues are not sufficient.

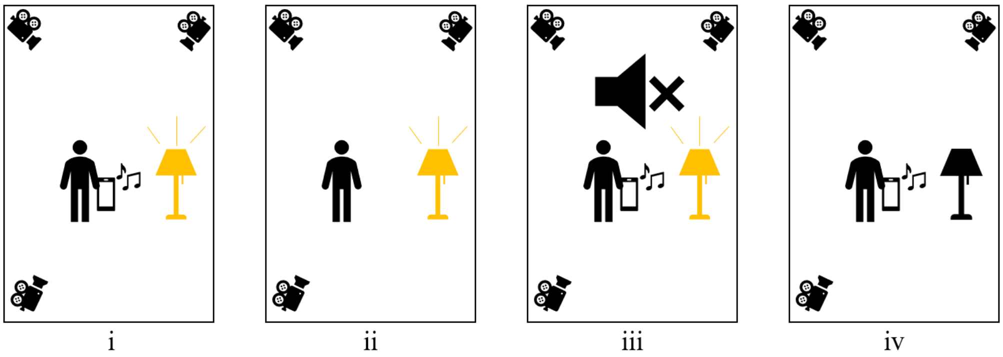

Figure 1. Different scenarios to test the synchronization and viewpoint score calculation.

1.2. Observations on the System

We mainly tested the system with the videos with scenarios described in Sec. 1.1. In addition, we also fed the system with few videos beyond the scenarios. By testing the system, we arrived at following observations.

A) Video Alignment Algorithm

In scenarios ii) and iii), the system failed to generate the output video. Those videos have insufficient or no audio information for the system to synchronize the videos. However, even there was almost no visual cue for computer to understand, the system generated the output video for the scenario iv). Therefore, we conclude that the system requires the synchronization step for at least to generate the output video, and the process uses audio cue.

B) Output Audio Source

We carefully listened to the audio of the output files of scenario i) and iv). We could hear that there was no transition between the audio signals that the volume of the music didn’t match the viewpoint that there were moments where the volume appeared small when the person was close to a chosen viewpoint at certain moment. We address that, successfully generating the output implies the audio signals across the cameras are distinctive enough to synchronize the videos. That is, the audio in all the videos have sufficiently informative and there is no need to translate between audio signals that can cause unnatural result.

C) Temporal Domain of Output Video

When the videos are fed into the system, we observed that the system generates output video whose temporal domain is the intersection of all the input videos. That is, no matter how long each video is, if the intersection across all the videos is short, the output will only have the intersection. We suppose this is because of the alignment is based on audio, which can be performed on the intersection only.

D) Aspect Ratio of Output

The videos generated by the system has specific resolution of 360×640. When the videos are fed into the system, the videos are resized to the resolution. It causes geometric distortion and causes unnatural appearance of the objects in the video. When the portrait (Height > Width) videos are fed into the system, the distortion gets worse and the objects from the viewpoint (considered to be ‘important’ by the system) looks very unnatural.

Among the observations, we address to alleviate the limitation of temporal domain of the output video as in Fig. 2. In the following section, we demonstrate the pipeline to generate better output video out of input videos with limited intersection.

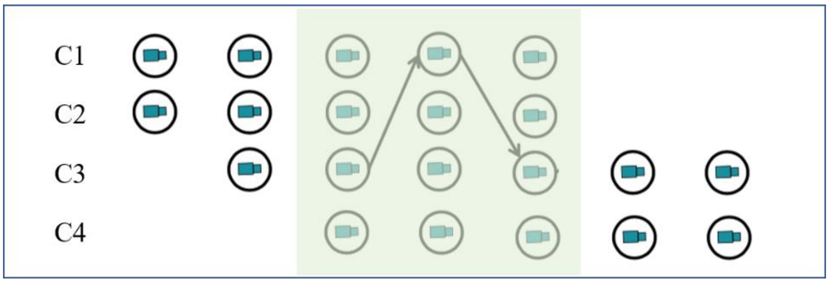

Figure 2. Possible temporal domain of the output video as shaded region.

2. Proposed Pipeline

To achieve better performance of the given system, we come up with few considerations of addition/adjustment of module(s). Since we are not accessible to the inner structure or code of the system, we propose a pre-processing pipeline to alleviate the limitation of limited temporal domain of output video, as in Fig. 3.

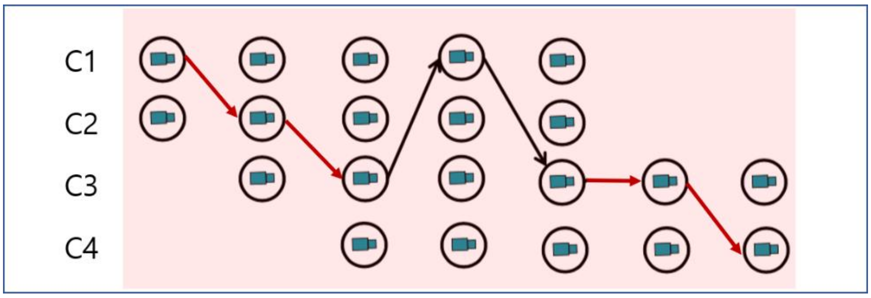

Figure 3. Expected output of applying our pre-process pipeline.

2.1. Overview

We collect the videos of a single event from social media, e.g., Instagram. Then we make use of audio alignment method to find the temporal offset of the input videos. Then we choose audio signals from different videos and concatenate few of them, that the concatenation hs the union of each audio signal in temporal domain. Then, based on the stitched audio, we insert the videos into corresponding time slot and padded in temporal axis.

2.2. Video Crawling

To show the effectiveness of our proposed pipeline, we first collected videos of a same event. To collect videos from online, we used an open source code from GitHub, called Instagram-scraper [1]. It is python-based code that takes hashtag and automatically download the materials that contains the input hashtag. The Instagram videos collected have fixed width of 480 pixels, and frame rate of 30fps.

2.3. Audio matching and Stitching

Even the given system uses audio signals to align the input videos, we used external audio alignment code [2] since we don't have the access inside the system and the output video only provides the sound of the intersection of the input videos. When this code is given with videos, it extracts audio signals and execute the matching, then returns the amount of time to be clipped or padded for the videos to be aligned. Using this time information and audio sampling rate, we can align and stitch the audio signals. We set up a strategy to minimize the number of transitions between audios to reduce the unnaturalness in audio. When the audio of the first video is finished, the audio from the one which started most recently is selected, and this process is repeated till the last video. The audio signal selection process is described in Fig. 2.

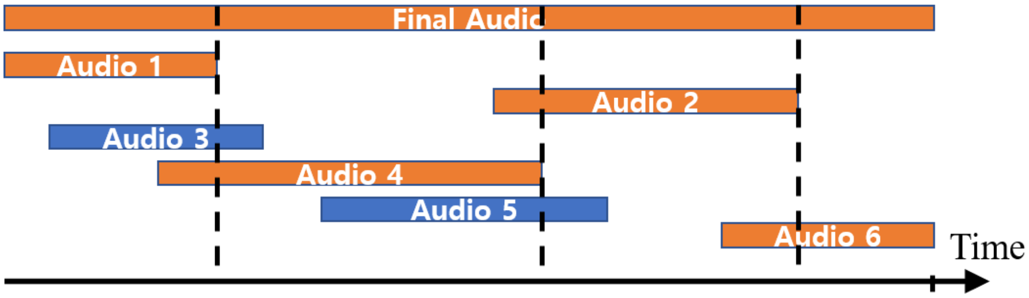

Figure 4. Illustration of our audio stitching process. Orange-shaded audios are selected and stitched.

2.4. Video Processing

Finally, using the time offset information obtained during the audio stitching process, we pad the videos temporarily in front and/or back that all the videos have length of the union of all the videos, as in Fig. 3. In addition, all the videos have same audio that is obtained with the process in Sec. 2.3. In such way, the given system is free from possible errors from the synchronization process. In this padding process, we assume that the system would not choose a scene that has no intensity level or motion at all, so that it will show select the ones that have visual information at the moment.

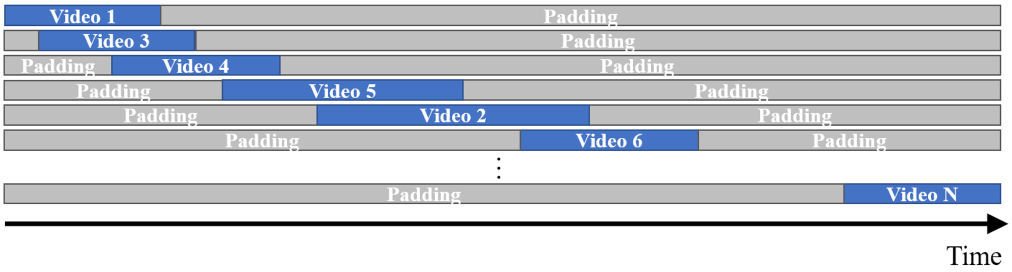

Figure 5. Illustration of our video padding process. Each video is padded into full duration.

3. Results

In this section, we analyze our proposed method by qualitative evaluations. We investigate the contribution of our method with respect to 1) the length of the output video, 2) the quality of generated audio, and 3) the quality of generated video. We use 14 input videos to generate the video that represents full event, all videos have different viewpoints of camera.

3.1. The comparison of the length of the output video

Tab.1 shows the comparison between the length of the output video generated by the conventional method and ours. Since the intersection of input videos does not cover the full duration of the event, the output video of the conventional method is only 4 seconds long. However, ***we expect*** that our method achieves generating full event by matching the length of all videos through padding. **(At the moment this report is written, the server for the given system is not working well, that we were not able to observe the result from our processed videos.)**

Method	Output Video Length
Conventional	4 seconds
Ours	

Table 1. The comparison of the length of the output video with 14 input videos.

3.2. The qualitative result with the audio stitching

Figure 6. shows the spectrogram and its signal peaks used to identify the audio and to match several videos. The spectrogram is shown with amplitude as a function of time and frequency. With the sequence of the audio signal, each column represents the strength of the signal at that particular frequency with respect to a certain time. Black dots represent the peaks of the magnitude of the signal and used to match several input videos. By comparing these groups of "constellations" between videos, we can match intersection of several spectrograms of input videos.

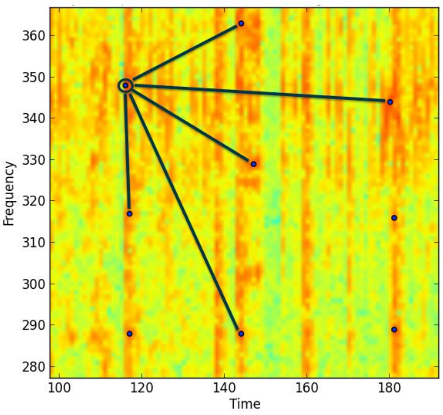

Figure 6. Spectrogram of the audio of videos 1. The x-axis represents time and the y-axis represents frequency of the audio signal. Black dots represent peaks of the spectrogram and we use these groups of "constellations" to match several videos.

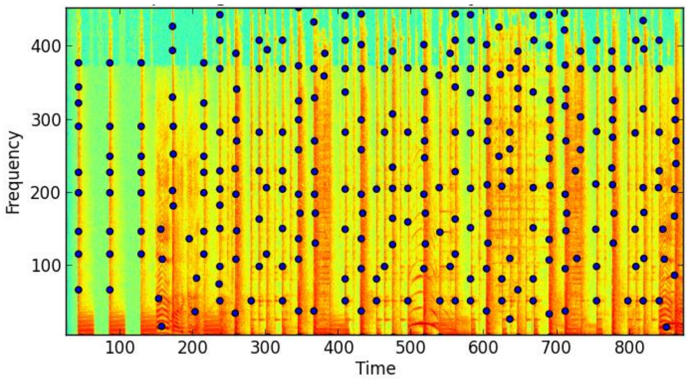

Figure 7. The spectrogram after matching all input videos.

The output spectrogram generated by matching all input videos is shown in Figure 7. By analyzing and comparing the constellations between videos, we can generate stitched audio sequence for full event.

3.3. The qualitative result with the output video

With the audio stitching and the video padding process, we ***expect to*** obtain a satisfactory output video which has long-time duration covering full event and well-selected best view for multiple input videos. **(At the moment this report is written, the server for the given system is not working well, that we were not able to observe the result from our processed videos.)**

4. Conclusion & Discussion

In this project, we analyzed a system which takes multiple videos a same event from different viewpoints. We address the problem that occur when input videos have only partial intersection, the domain of the output video is limited to the intersection. In order to solve this problem, we stitch audio signals from different videos to generate unified soundtrack, and temporarily pad the input videos so that they have same duration. By this processing pipeline, output video is generated from the union of the input videos. While our method enables to utilize entire parts of input videos, it also has some defects. One is that the stitched audio sounds unnatural at transitions. Another defect is that quality of the stitching process is highly dependent to audio stitching algorithm we used [2].

In addition, we came up with an idea to resolve the limitation of 1.2.A, that the system cannot process the videos without sufficient audio cues for alignment. Assuming that the videos contain same visual cues, we could apply Visual-to-Sound technique [3] to generate sounds for the videos without audio information. However, it is hard to expect such generative method generates consistent sound across all the viewpoints and this idea is not likely to be realizable.

To resolve the aspect ratio issue addressed in 1.2.D, we can apply simple pre-processing to the input videos. We can crop or zero-pad the input video frames into desired resolution or aspect ratio, then feed them to the network. Unless the contents are harmed, cropping process seems to be more desirable since the transitions between differently padded videos will distractive.

5. References

[1] "Instagram-scraper," Online, https://github.com/rarcega/instagram-scraper

[2] "Align-videos-by-sound," Online, https://github.com/jeorgen/align-videos-by-sound

[3] Y. Zhou et al., "Visual to Sound: Generating Natural Sound for Videos in the Wild," arxiv, 2017.