Tampere University

Einari Vaaras, Mehmet Aydin and Kalle Lahtinen

SIGNAL PROCESSING INNOVATION PROJECT

Automated spatiotemporal annotation of sound objects in a scene

Final Report

Faculty of Information Technology and Communication Sciences

Clients: Archontis Politis and Tuomas Virtanen

May 2021

ABSTRACT

Einari Vaaras, Mehmet Aydin and Kalle Lahtinen: Signal Processing Innovation Project
Final Report
Tampere University
May 2021

This report describes the goals, work and results of the Signal Processing Innovation Project course implemented in the spring of 2021 at Tampere University.

Annotating spatio-temporal audio data for machine learning based models is considered a laborious task with a lot of uncertainties especially in the evaluation of the spatial dimension. The goal of the project was to research the possibility of using visual detections from 360-degree video recordings in automatically creating annotated training data for audio model training. The project consisted of researching different detection and tracking methods available for 360-video data, learning about spatial audio processing methods and recording test scenes with a 360-video camera and a microphone array. Two different data processing pipelines combining the visual detections made from the video recordings with the audio recording are proposed. The proposed pipelines are based on framewise video detections from four stereographic subprojections using the YOLOv4 object detector and either audio powermap processing or audio beamforming for infering audio activity. Both of these pipelines provide training data with sound source location and activity included as an output.

:

CONTENTS

1. Introduction 1
2. Summary of the Project 2
   2.1 Project background 2
   2.2 Project objectives, deliverables and planned timetable 2
   2.3 Project organization 2
3. Realized Project Implementation 4
   3.1 Implementation Steps 4
   3.2 Meetings, Week Reports and Inspections 7
   3.3 Deliverables 8
   3.4 Timetable and Workload 8
   3.5 Budget 8
   3.6 Problems, Delays and Changes in Project Organization and Plans 9
   3.7 Lessons learned 9
   3.8 Future Development Needs 9
4. Project Results and Conclusions 11
5. Comments and Opinions on the Course 12
References 13

1. INTRODUCTION

This project was the course work for a Tampere University course 'Signal Processing Innovation Project' designed for students majoring in signal processing and machine learning. The course was completed during the spring of the year 2021. The students working on the project were Einari Vaaras, Mehmet Aydin and Kalle Lahtinen. The goal of the project was to study the possibilities for automatically annotating spatial and temporal dimensions of sound objects detected in an audiostream with the help of a 360-degree video recording. The premise was that the objects are over two meters away from the the camera, so the project involves limited elevation angles. Another basic element was that including multiple sound sources is relevant, but there is no need to go to extremes, such as two similar objects being very close to each other.

The need for such a tool rises from the research of spatial audio in which data intensive modelling methods are heavily used. The goal of the project was to provide a proof-of-concept application which would improve the quality of spatial audio data used in the training of such models and reduce manual labour related to the annotation of the data. The project was implemented using open-source tools. The main tool for development was Python and available signal processing and machine learning libraries. In addition to Python, Matlab-scripts provided by the client were also used for the audio processing. The resulting application and its source code are shared as open-source (`https://github.com/ktlhtn/AutoSTAnnot`) for further research use under the Aladdin Free Public License.

The project proceeded according to the project plan made at the very beginning of the course. The end result of the project work was two different proof-of-concept processing pipelines that combine detections made from 360-video recordings with spatial audio recordings from the same scene. The pipeline outputs provide annotated spatio-temporal audio activity data that could be used for model training. The training data created with the pipelines were not yet tested in actual model training, but the findings from this project indicate that the basic method could be useful and the topic should be further researched.

2. SUMMARY OF THE PROJECT

2.1 Project background

The project aim was to research and develop a set of tools for automatically annotating spatial audio data with the use of visual detections made from 360-degree video recordings.

2.2 Project objectives, deliverables and planned timetable

The overall goal of the project was to develop a proof-of-concept for providing spatiotemporal labels for audio events using a 360-degree camera and a microphone array. In addition to the course-specific deliverables such as the project plan report, the final report and the final presentation, the main deliverable of the project was agreed with the clients to be a GitHub repository containing the source codes that were created for the components of the project. Additionally, a thorough documentation of the source codes was agreed to be delivered on the GitHub repository. Practically the only timetable for the project was that some kind of functioning pipeline would be finished by the end of the course. On a weekly basis, the tasks for the next week were agreed on together with the clients on a Teams meeting.

2.3 Project organization

The project members were Einari Vaaras, Mehmet Aydin and Kalle Lahtinen. The open tasks related to the project were discussed on a daily basis in a project-specific Telegram channel. Before the project, there were no predefined roles for the project members, and the aim was to divide weekly tasks based on open discussion and equal contribution. During the project, each group member specialized in different categories of the project. Einari handled parts related to audio powermaps, beamforming, bounding box cleaning and mapping class labels. Mehmet specialized in experimenting with the recording devices such as providing test-scenes-to-be-experimented with on a weekly basis. Additionally, he studied about tracking methods and their implementations, especially Deep SORT [1]. Kalle took care of the video object detection pipeline, as well as defining a standardized CSV file output and producing audio event detections based on audio powermaps. In

problematic situations such as code debugging, all group members participated in helping the group. In addition, course-related activities such as writing reports and planning presentations were carried out by all group members. Furthermore, all group members participated in the final recordings for the project, which are further discussed at the end of Section 3.1.

3. REALIZED PROJECT IMPLEMENTATION

3.1 Implementation Steps

At the end of the project, there were two alternative pipelines. These pipelines are depicted in Figures 3.1 and 3.2.

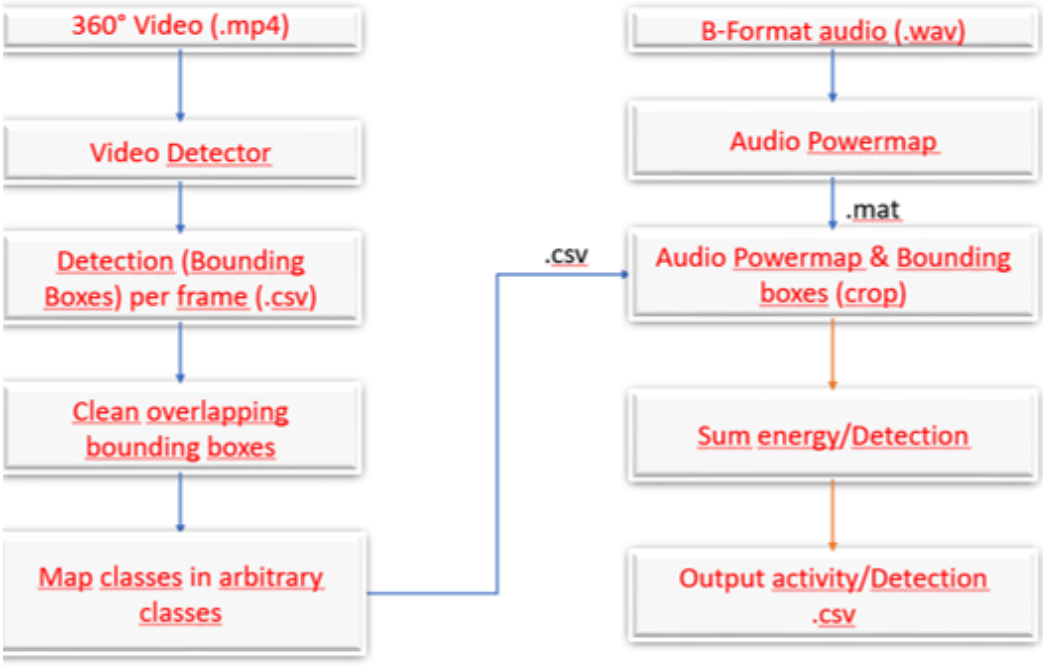

Figure 3.1. *First variant of the project pipeline.*

In the first variant of the pipeline (Figure 3.1), the 360-degree video is first used as an input to a video detector that processes it frame by frame. The video object detection is based on a widely used pre-trained video-detection network called YOLOv4 [2]). However, as 360-degree equirectangular video frames are significantly curved and distorted, some framewise processing was required before the detector could be used. Each frames were projected to four stereographic sub-projections (or sub-images). For each of these four sub-images the visual detections were made individually after which the sub-images were used to create the full equirectangular frame with the detection coordinates transformed to the coordinates of the 360-frame. The frame projections were done with the source

code available in a Github-project [3] which again was based on a paper by Yang et al. [4].

Then, when the video detections have been acquired, it can be observed that there are multiple overlapping bounding boxes for detections of the same class instances. Therefore, a function which cleans overlapping bounding boxes was implemented. The basic idea of the iterative bounding box cleaning function is the following:

1. If there are same classes present in a video frame, then proceed to part 2.
2. If the detected bounding boxes of the same object classes are overlapping more than a predefined threshold, then proceed to part 3. The overlap is determined by the intersection over union.
3. Remove the bounding box with the least confidence as determined by the visual object detector.

After cleaning the CSV file of overlapping bounding boxes, it is possible to map any given class or classes from the object detector into a new arbitrary class. For example, all types of vehicle object detections can be mapped into a new class calld ‘vehicle’. For this, a class mapping function was created.

Up until now, all descriptions have focused on the video detection side. For the B-format audio (4 channels) obtained from the microphone array, a function provided by one of the clients (Archontis Politis) converts the audio into a powermap representation using the MUSIC (MUltiple SIgnal Classification) algorithm. The powermap provides a framewise mapping of sound energy in the 3 dimensional space captured by the microphone array. For each recording the framerate of the video and the audio based powermaps were the same. The information from the visual detections were used to crop the corresponding powermap frame so that all sound energy inside the detection bounding boxes are kept unchanged and everything else is set to zero. The audio activity for a given label at a certain direction is the inferred by summing up the cropped powermap values together. If the powermap energy exceeds a set threshold parameter, the system infers that there is audio activity caused by that visual detection in the direction of the bounding box center coordinates. This is done for each frame and for each visual detection one at a time. As an additional information the bounding box center coordinates are also converted from cartesian coordinates to spherical coordinates, as spherical coordinates is more often used in spatial audio processing.

In the second variant of the pipeline (Figure 3.2), the initial parts of the video processing side are the same as in the first variant of the pipeline. Now, the B-format audio is not converted into an audio powermap. Instead, the audio is used as an input to a *beamformer*, whose code was also provided by one of the clients (Archontis Politis). The beamformer basically “listens" to a given direction. In our case, this direction should be the direction

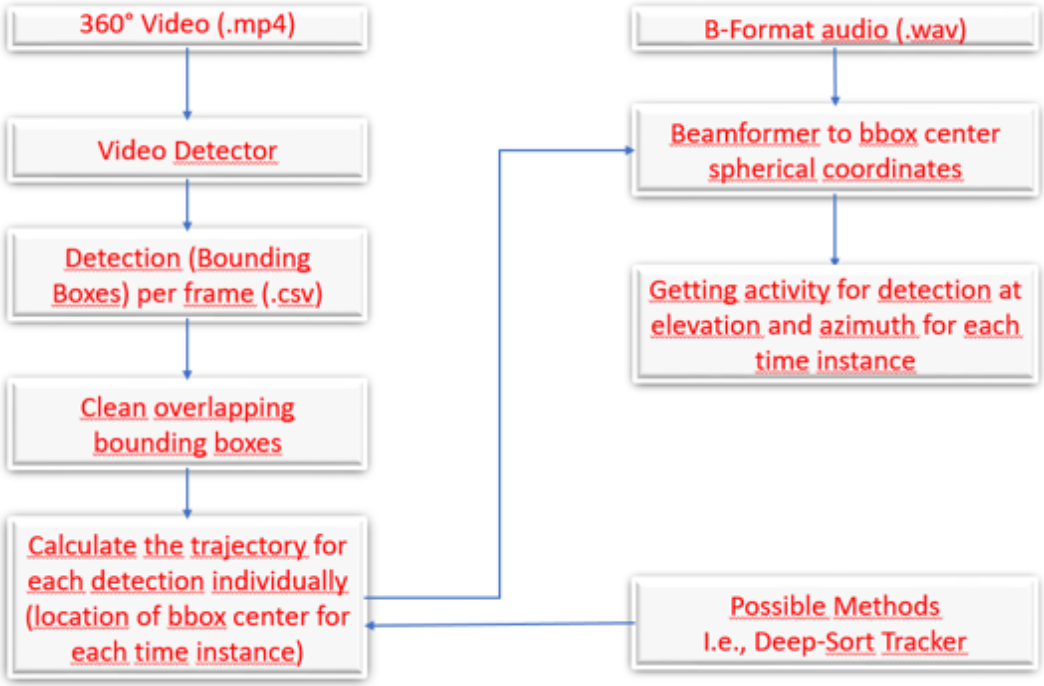

Figure 3.2. *Second variant of the project pipeline.*

of a given object.

What turned out to be problematic for the beamforming was that it is difficult to determine associations between objects in a given scene, e.g. is a given object instance one of the object instances of the same class in the previous video frame. Therefore, two alternate cases were implemented: 1) Audio segments of multiple video frames, and 2) Audio segments of a single video frame are used as an input to the beamformer.

When inputting audio segments corresponding to multiple video frames, the pre-processing code was simplified so that only one object could be present in a given video scene. The temporal trajectories of this given object were then determined so that if the object was appearing for more than four consecutive video frames, then its temporal trajectory was taken into account. If the object did not appear for more than four consecutive video frames, then its temporal trajectory was discarded. After obtaining the temporal trajectories of the given object in a scene, the sound segments and the frame-level bounding box centers (converted into azimuth and elevation in radians) that correspond to a given temporal trajectory were given for the beamformer to produce a spectrogram-like energy representation for each audio frame of the object. Finally, if the sum of the frequency bin-level energies exceeded a predetermined threshold, the audio frame was said to contain audio activity. Furthermore, since the audio frames had a different framerate than the video had, the audio activity detections were determined based on the sums of the bin-level energies that were interpolated into a 10-fps framerate.

In the other variant where the audio segments of single video frames being used as an input to the beamformer, all objects within a video frame were first listed. Then, for each object detection in a video frame, the location (azimuth and elevation of bounding box center) and the corresponding audio segment (100-ms segment for 10 fps video) were used as an input for the beamformer. For the spectrogram-like energy representation produced by the beamformer, the total energy of all frequency bins was summed. If this sum exceeded a predefined threshold, then the visual object was determined to have audio activity in the given video frame.

To get more intelligent temporal trajectories of visual objects, visual object tracking methods should be applied. An example of a visual object tracker is Deep SORT [1]. Deep SORT is one of the simplest algorithms used in object tracking. It is used for tracking multiple objects in real-time applications. Like the SORT algorithm, the Deep SORT algorithm uses the Kalman Filter method to predict the location of objects in the next image. Deep SORT and SORT algorithms are separated from each other by the method they use when associating objects. A convolutional neural network (CNN) is designed for object classification to be used in the Deep SORT algorithm. The convolutional neural network is trained until high accuracy is achieved. Thanks to CNN, the most distinctive feature of the object to be classified, and the most distinctive feature that distinguishes the object from other objects, is tried to be determined. In the last layer of the CNN structure, the classification of the objects is done. This classification process is done according to a vector representing the object. In the Deep SORT algorithm, a vector is obtained by passing each detected object through the neural network and using these vectors to associate the two objects. This vector is called the "appearance feature vector" According to the SORT algorithm, the Deep SORT algorithm is more successful in object association in cases such as occlusions since it is more specific and distinctive features of the object are examined to associate an object with another object.

To test the methods during development, small test scenes were constantly recorded during the project using the 360-video camera and the microphone array. At the end of the project, all three group members gathered at the university to produce carefully planned recordings to demonstrate and highlight functioning as well as non-functioning parts of the pipeline.

3.2 Meetings, Week Reports and Inspections

During the project, almost every Friday at 14:00 a MS Teams meeting was organized together with the clients to discuss updates about the project from the past week, and to set goals for the following week. Additionally, communication with the clients was carried out via MS Teams and email.

3.3 Deliverables

The main deliverables of the present project were the source codes and their thorough documentation created during the project which are provided in a GitHub repository. Additionally, the present final report is the main deliverable course-wise.

3.4 Timetable and Workload

As was originally planned, the project proceeded according to what was agreed on each weekly Teams meeting with the clients, with the aim to complete the given tasks during the following week. After each Teams meeting, the group members discussed about the division of the workload so that each member would have approximately the same workload as other group members.

There was no specific weekly timetable planned for the project before the start of the project. The main timetable was that at least one functioning version of the task pipeline should be finished by the end of the course, which was achieved successfully. Furthermore, we were even able to provide an alternate functioning version of the pipeline.

3.5 Budget

For achieving the project goals, approximately the time planned in the initial project plan was used to the project. That is, approximately 133.3 working hours for each group member. By assuming that all group members would be recently graduated M. Sc. (Tech.) workers with the recommended starting salary by TEK (the largest organization for academic engineers in Finland) which is 3980€/month, the project would cost approximately 9256€ altogether.

The two clients used approximately 30 minutes for each remote meeting which occurred almost every Friday (for calculations, we can assume that the remote meeting occurred every Friday). In addition, if it is estimated that the clients both use approximately 30 minutes for the project each week outside the meeting time (such as reading and answering to messages on MS Teams), this gives an additional two working hours each week for the clients. Furthermore, one of the clients used time to provide code for the project group. If we assume that this contribution would take one hour on average for each week, we would have that altogether the clients used approximately , this leads to approximately 42 working hours for the clients. This is significantly over the initially estimated 28 hours (50% more). If an estimate of the salaries of the clients would be e.g. 5000€ a month, then their input to the project would cost approximately 1221€.

3.6 Problems, Delays and Changes in Project Organization and Plans

During the project, there were no major issues hindering the progress of the project. However, out of the original project plan, testing the implementation against manual annotation was not put into practice. This was due to the fact that instead of only finishing one possible version for the project pipeline (audio powermaps), another version of the pipeline (beamforming) was also implemented. This additional version was not planned during the beginning of the project, and it was proposed near the end of the project by the clients.

3.7 Lessons learned

What was noticed during the project was that without a visual object tracking method, it is extremely difficult to define correspondences of objects across video frames. In addition, although there are multiple different types of projections (e.g. stereographic or perspective projection) to convert a curved equirectangular image into a less curved version, it was observed that practically all of these projections have their pros and cons, and no projection is able to perfectly convert an equirectangular image into an "ordinary" image. For example with the stereographic projection that was used in the present experiments, it was observed that objects that are very close to the camera (and therefore very curved) are not well detected by the video object detector, even though a stereographic projection is applied to the equirectangular image before using the video detector. Furthermore, what stood out as new information was that it is not possible to get an accurate sound localization using only four microphones, although four microphones might sound like a lot for conventional applications.

In addition to lessons regarding object detection and signal processing was the importance of licesing and especially carefully studying the licenses of any 3rd party component. The original developers of the multiprojection-yolo project that was used as a basis for the video detection side initially did not have a licensing file added to the repository in Github. This could have caused problems, as the default licensing principle in cases where there is no license in the project, Github states that the code can not be used, copied or distributed in any way. Luckily the original developers could be contacted via email and they were happy to release their project undert the MIT-license.

3.8 Future Development Needs

There are several ways to improve the presented implementations. First, the implementation should be tested more vigorously. For example, now there were only five well-

designed indoor scenes recorded, whereas a greater number of recordings, both indoors and outdoors, should be recorded to better highlight the pros and cons of the present pipeline. Also, the outputs presented implementations should be tested in actual sound event detector training to showcase that how do the automatically generated labels perform in real-life use cases.

For improving the video detector side, the present pre-trained video detector is only able to detect a little over 80 visual objects, out of which the majority (e.g. pizza, frisbee, parking meter etc.) are not useful at all for representing sound-making objects. The video detector could further be developed by only training it with objects that are useful for the present task. In addition, it was observed in the test scenes that if some object is very close to the camera, it is not detected from the equirectangular image, although the video processing pipeline handles curved visual objects on some level. Therefore, either the video detector should also be trained with images that are warped a little, or the recorded scenes should be designed more carefully so that objects are not very close to the camera. Furthermore, even though a 4k resolution that the Ricoh Theta V camera produces is a rather good video resolution for conventional video, in fact a 4k resolution is considered to be a small resolution in 360-degree video. To better detect objects (especially objects that are far away from the camera), a camera with a better resolution should be used.

To better improve the audio processing side, both the audio powermap and the beamforming could be improved by trying out the system with a microphone array with a better spatial resolution, i.e. more microphones should be used (only four microphones were used in the present experiments). Also, the present audio processing implementations require that sound-making objects should not be very close to each other. This could also be improved by using microphone arrays with more that four microphones. To further improve the beamforming method, video tracking methods should be used together with the object detector to intelligently produce time arcs of visual objects in a scene.

Finally, the present proposed pipelines work piece-by-piece, but are not designed to function in an end-to-end manner. To make producing audio event labels easier, the proposed pipelines could be made end-to-end so that the final output CSV could be produced e.g. by inputting an MP4 video file with B-format audio into a program together with some configuration file which determines the details about the way the user wants to produce the audio event labels.

4. PROJECT RESULTS AND CONCLUSIONS

The main goal of the project was to get a working pipeline for automatically generating spatiotemporal labels for audio events, and the project group managed to get two functioning pipelines for the task successfully. As an outcome, a GitHub repository containing all source codes that were created during the project was created together with a thorough documentation of how the provided codes can be used for the task. Overall, the project provided valuable insight into the topics of object detection for 360-degree videos and spatial sound recognition for both the group members and the clients. However, as listed in Section 3.8, there are multiple areas of the project that can be vastly improved. Perhaps one of the main outcomes of the project was to gain knowledge for the clients of what is currently working and what is not, and what are the limitations of the current pipelines and the equipment that was used.

5. COMMENTS AND OPINIONS ON THE COURSE

Overall, the project was very educative. What was a welcome observation was that none of the suggested topics on the course were fully in the comfort zone of the group members. All of the given topics included plenty of areas of signal processing and machine learning that were completely new to the group members, which forced the participants to choose a topic that contained lots of uncertainty. To our experience, this turned out to be a clear advantage, since the present project taught each group member a lot about different areas that are completely new to the group.

In addition to the challenges that implementing the project provided, the course gave a good insight on project management and teamwork coordination. The course also gave some hands-on experience on how some things might be done in working life, such as planning the timetable and estimating the budget of the project.

REFERENCES

[1] Wojke, N., Bewley, A. and Paulus, D. Simple Online and Realtime Tracking with a Deep Association Metric. *arXiv preprint arXiv: 1703.07402* (2017).

[2] Bochkovskiy, A., Wang, C.-Y. and Liao, H.-Y. M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv preprint arXiv: 2004.10934* (2020).

[3] *MP-Yolo Github-project*. May 5, 2021. URL: https://github.com/keevin60907/mp-YOLO (visited on 02/05/2021).

[4] Yang, W., Qian, Y., Cricri, F., Fan, L. and Kamarainen, J.-K. Object Detection in Equirectangular Panorama. *arXiv preprint arXiv: 1805.08009* (2018).