Automated Killstreak Extraction in CS:GO Tournaments

Michael Wutti
Alpen-Adria-Universität Klagenfurt, Austria
mwutti@edu.aau.at

ABSTRACT

This paper describes a system for automated killstreak extraction in CS:GO tournaments. It uses a metadata and content-based approach for identification of the killstreaks. Inaccurate timestamps in the provided metadata of the tournament resulted in almost unusable detection results. Therefore, a content-based technique has been developed to extract the correct highlight videos. The results are merged into a summary video with a multi-view perspective of the event stream, the actor's view and the views of the victims. Due to the content-based approach, the multi-view video could be synchronised nearly frame accurate.

1 INTRODUCTION

E-sports events such as the CS:GO World Championship 2018 in Katowice are becoming increasingly popular. One of the key challenges of such an event is to summarise it in an exciting and captivating manner. E-sports events are rich in multimedia data such as video and audio streams, player details, commentator descriptions and comments from viewers. One of the tasks in the MediaEval 2018 Workshop [2] is to find a way how E-sports matches ramp up, evolve and play out over time. This paper shows an approach to cover a subset of all the different aspects and possibilities for summarising E-sports events. The developed software is written in Python and uses a couple of image processing and machine learning frameworks such as OpenCV [1] and scikit-learn [3] to automatically extract killstreaks from the provided videos. The final goal in the GameStory Task of MediaEval 2018 is to summarise the last match of the tournament (approx. one hour) in less or equal than five minutes. Every team is allowed to submit three different versions of the highlight video. In the next sections the process of summarising a CS:GO match by extracting the most valuable killstreaks is presented.

2 META DATA PROCESSING

Two different types of metadata are provided by the task organisers. The stream metadata consists of information about the video streams including timestamps and duration of the matches. The game log contains a detailed listing of all events such as purchases, kills and the winning team for each round. Unfortunately, the timestamps in the stream metadata may differ up to 40 seconds from the timestamps in the game log which is a typical problem in multimedia systems. In order to use the information in the game log, it is therefore necessary to find out the exact time of an event. Based on the game log a model has built which contains additional information about the game score for each round and keeps track of the overall game state. Additionally, a killstreak list has been generated

MediaEval'18, 29-31 October 2018, Sophia Antipolis, France

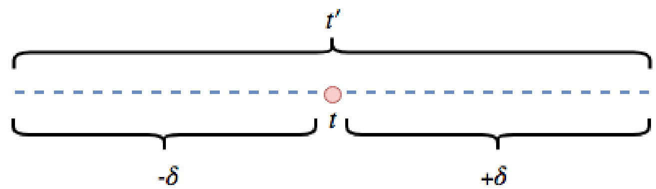

Figure 1: Displacement of killstreak start: t... killstreak begin extracted from the game log, δ... maximum displacement from the actual killstreak begin, t'... possible range of correct killstreak begin

by processing the game log. This list is used to identify all players who were involved in a killstreak (actor and victims) and to store the start time and the duration of the killstreak.

3 EXTRACTING KILLSTREAKS

With the generated model all killstreaks with a maximum duration of 20 seconds and a minimum killstreak length of three players were identified as candidates for the summary. As seen in Figure 1 the main difficulty in extracting the killstreaks is to identify the start time of the killstreak. The begin of a killstreak t extracted from the game log may differ up to $\delta = 40$ seconds from the timestamps in the stream metadata. This means that the correct start point t' could be in the range of:

$$t - \delta \leq t' \leq t + \delta$$

To find the correct timestamp, an additional video processing step has been implemented. First, the beginning of the round in which the killstreak was occurred has been identified. Second, based on the round begin and the information in the model the beginning of the killstreak has been found.

3.1 Detecting Round Start

To determine the correct start of a killstreak, the model retrieves the number of kills that were done before the killstreak has started. Based on this number k the timestamp of the first kill from the killstreak can be determined. But due to the potential displacements of the timestamps, there can be two consecutive rounds in the search window $t - \delta \leq t' \leq t + \delta$ in which an equal number of kills were occurred. Therefore, it is necessary to first identify the round begin of a killstreak. As an indication for the correct round start, the difference between the timestamps of the match begin and the (target kill + δ) is added to the given video position of the target match. Starting from that point the match begin is being searched.

As seen in Figure 2 the match score is displayed in the upper part of the video. Two Regions of Interest (ROI) are extracted where the score numbers are displayed. A Support Vector Machine (SVM) is used to classify the ROIs. The SVM was trained with 400 samples of

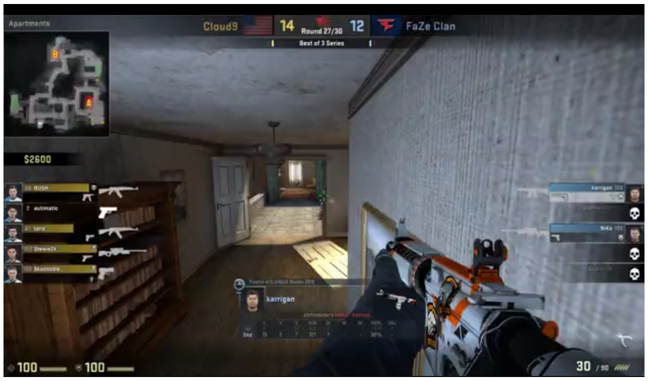

Figure 2: Gameplay screenshot with match score and skull indication of achieved kills.

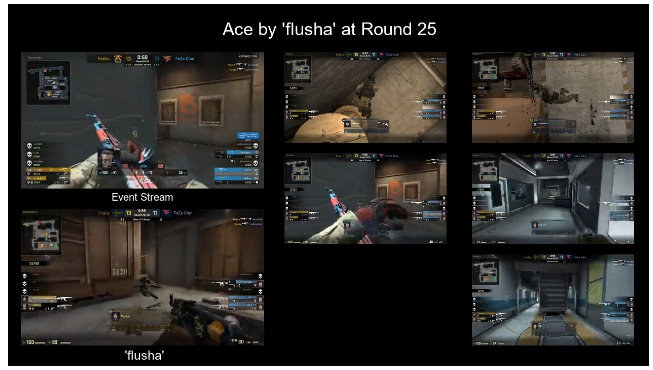

Figure 3: Screenshot of the multi-view killstreak summary.

each number from 0-21 which where extracted beforehand from the training set. Every frame in an extended search window with $\delta = 1$ minute is processed by the SVM and after an amount of 60 matches with the target score, the timestamp of the round start t_r is detected. The decision for choosing a SVM for the number detection problem is based on the fast computation speed and satisfying precision that was achieved during development.

3.2 Detecting Killstreak Start

A pattern matching approach is used for the detection of the killstreak begin. Given the information from the model, that n kills were achieved before the killstreak has begun, the timestamp t_k where $(n + 1)$ kills were achieved is being searched. As seen in Figure 2 a kill is indicated with a skull at the left or right side of the screen. For that reason ten ROIs where a skull could possibly appear have been extracted and compared with a reference picture of the skull. The comparison was done with the L1-norm of the normalised reference and ROI values. If the norm has fallen below a threshold value the kill counter is increased. At the moment the kill counter reaches $(n + 1)$, t_k is reached and can be used as the start point for extracting the video of the killstreak.

3.3 Extracting Videos

The technique described above has first been applied to the event stream. Later it has been used for extracting the killstreaks from the player streams. To achieve a better control of the target video length and a better overall viewing experience, a clipping-offset between 5 and 20 seconds is applied before t_k and after the killstreak end. Because every timestamp t_k is a result of a content-based video processing step, the resulting video set of the player views and the event stream is synchronised nearly frame accurate. Based on the correct timestamps and the duration of the killstreaks, the videos are extracted using ffmpeg and the multi-view summary seen in Figure 3 is created with OpenCV.

4 RESULTS AND DISCUSSION

Three different summaries were created from the extracted videos. Every video consists of nine killstreaks, ordered by their corresponding round numbers. Video one and two show the synchronised event view and killstreak actor view. As seen in Figure 3 the last video is a multi-view summary and is made from the event view, actor view, and the views of all victims. The videos where watches by two to four reviewers from MediaEval and evaluated based on five criteria: match summary, entertainment value, flow and peak of a good story, innovation and portability for other games. Some of the shortcomings of the reviewers should now be briefly addressed.

The summary covers only the killstreaks greater than three which were achieved during the match. This seems like a quick and easy approach. But if the timestamps in the metadata files can differ up to 40 seconds from the actual time, the provided information is useless. Because the compensation of the timestamp difference was an unexpected time-consuming task, many ideas like the integration of overtime events and audio-based analysis could not be done.

Another aspect which is not covered by the summary is the impact of money and the player's purchases at round begin. An integration and analysis of this data could have produced additional highlight videos. For example, one could investigate if a team is playing with a weak load-out against a team with strong equipment and still wins the round. In addition, sneak and melee kills could be easily integrated by analysing the metadata. Nevertheless the metadata timestamps of such events must be synchronised with the actual event times. Other aspects like pitfalls and mistakes made by other players are definitely interesting and would make the summary more exciting.

The algorithm presented in this paper is able to automatically detect and extract sequences of killstreaks from the whole tournament data. Nevertheless, killstreaks are just a subset of events which could be used for an exciting and captivating game story. In future work, this algorithm will be extended and adapted according to the feedback of the reviewers.

ACKNOWLEDGMENTS

Special thanks to Mathias Lux who has contributed as a supervisor and coordinator of this project.

REFERENCES

[1] G. Bradski. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).

[2] Mathias Lux, Michael Riegler, Duc-Tien Dang-Nguyen, Marcus Larson, Martin Potthast, and Pål Halvorsen. 2018. GameStory Task at MediaEval 2018. In *Working Notes Proceedings of the MediaEval 2018 Workshop*.

[3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.