

Extracting Gameplay Information from a Projection of 3D Modelled Objects using OpenCV

David Kang, Irene Wachirawutthichai, Matthew Munson, and Sana Suse

Github repo: <https://github.com/DaviidK/mercy-detector>

Background:

Existing research has demonstrated that it is possible to parse visual data from a video feed of a game and analyze different aspects of gameplay. Static 2D elements such as a GUI used to represent health or energy have been studied in-depth, and no longer represent a particularly challenging task for computer vision projects. 3D modelled objects pose a similar challenge to identifying an object in real-life.

Overwatch is a first-person shooter developed by Blizzard Entertainment. The game allows players to select different characters and fight to complete specific tasks. These characters (specifically called "heroes") each have unique weapons and abilities, which are displayed using specific models and animations.

Brief Description:

Our goal is to create a project that investigates different techniques for interpreting a video feed of Overwatch gameplay. Our project will extract data from this feed and use it to understand certain aspects of the current game state.

We will focus on the 3D object held by the player's hero, which will move slightly with hero movement and execute different animations depending on the inputs given by the player. Using this information, we will determine two aspects of gameplay:

- 1) Which hero is currently being played
- 2) What action is the held weapon performing at a given point in time



In its final stage, we want our project to receive a set of frames of Overwatch gameplay and use them to output both the hero being played and their current action. We hope for this system to be performant enough to operate in real-time, which would allow for the analysis to be applied to a live video stream.

Input and Output:

Our system's input will consist of a video file depicting Overwatch gameplay. It is assumed that this video contains only Overwatch gameplay and has a consistent resolution. In a real-world application this could be swapped with a live video stream.

Our system's output will consist of a text stream describing the hero that is currently playing and the action that their weapon is performing. We intend to display this text stream in a terminal along with the video stream so viewers can compare the actual gameplay with the system's interpretation.

Project Evaluation:

Both tasks for our project will be evaluated using similar methods. For the hero detection, we will measure the accuracy of our system in detecting the correct hero being played. A test video will be manually created where each frame has a corresponding ground truth of the hero that is being played. This will be compared with the output of our system to determine accuracy.

For detecting the action of the weapon, we will once again create a test video, this time labeling each frame with the action being performed. Our system's output will be compared against these labels to determine the accuracy of action detection.

Schedule of Work

Milestone 0: Proposal - May 3rd

- Submit proposal

Milestone 1: Background Research - May 8th

- Research object detection in OpenCV and how to create reference models for 3D objects
- Investigate how to best preprocess images for hero and weapon action detection
- Start implementation for object detection in both hero and weapon action detection programs
- Create and format documents for the final paper

Milestone 2: Prototype implementation - May 17th

- Research basic video feeds and video processing using OpenCV
- Investigate what fuzzy logic may be needed to remember state over multiple frames of a video feed
- Write the introduction and prior work sections of the final paper
- Prototype Implementation to be showcased in the design presentation:
 - A prototype that can detect a single hero from an input image
 - A prototype that can detect a single weapon action from an input image

Milestone 3: Early Implementation - May 24th

- Create test videos for hero and weapon action detection
- Implement a system for hero and weapon action detection that can distinguish between two heroes and two weapon actions from one single image
- Start implementing the fuzzy logic system to maintain the state of the detected object in the video
- Prepare for research paper presentation

- Write implementation section of the final paper

Milestone 4: Late Implementation - May 31st

- Implement a system that can detect heroes and weapon action over time from a single video file, with the logic to maintain the state between frames
- Create test videos for hero and weapon action detection
- Test detection system with test videos and collect success metrics, and create plots for the paper
- Start writing the results section of the final paper

Milestone 5: Evaluation and Tuning - June 7th

- Evaluate the hero detection system and weapon action detection system with created test videos
- Tune parameters in both detection systems to produce more accurate results
- Create video demonstration
- Write the remaining sections of the final paper and proofread for submission

Delegation of Work:

The project will be divided into two main sections: hero detection and weapon action detection. We plan to work as pairs to develop the specific systems which will conduct each action. However we will work together as a larger group to create the infrastructure that will process the video input and eventually use the two detection systems to produce output.

The hero detection team will consist of Sana and David. The weapon action detection team will consist of Irene and Matthew. It is anticipated that the hero detection team's system will focus on the full-size UI as the heroes make up larger shapes. The weapon action detection team will focus on smaller areas and differences in color as weapon actions involve small shapes with noticeable differences in color.

We expect that both teams will progress at roughly the same pace, and that learning, implementation, evaluation, and tuning will occur during the same periods of time.

Publically Available Resources to Use:

- OpenCV for image processing
- Overwatch for image sourcing
- Git/Github
- Visual Studio 2019 Community

Bibliography:

- K. Kozłowski, M. Korytkowski, and D. Szajerman, "[Visual Analysis of Computer Game Output Video Stream for Gameplay Metrics](#)," in *Computational Science – ICCS 2020*, vol. 12141, V. V. Krzhizhanovskaya, G. Závodszky, M. H. Lees, J. J. Dongarra, P. M. A. Sloot, S. Brissos, and J. Teixeira, Eds. Cham: Springer International Publishing, 2020, pp. 538–552.