

Facial Landmark Annotation and Fuzzy Model to Detect Players' Focus

Alisson Steffens Henrique[†]
Mestrado em Computação Aplicada
Univali
Itajaí, Brasil
ali.steffens@gmail.com

Esteban Walter Gonzalez Clua
Medialab
Universidade Federal Fluminense
Rio de Janeiro, Brasil
esteban@ic.uff.br

Rodrigo Lyra
Laboratory of Applied Intelligence
Univali
Itajaí, Brasil
rlyra@univali.br

Anita Maria da Rocha Fernandes
Laboratory of Applied Intelligence
Univali
Florianópolis, Brasil
anita.fernandes@univali.br

Rudimar Luis Scaranto Dazzi
Laboratory of Applied Intelligence
Univali
Itajaí, Brasil
rudimar@univali.br

ABSTRACT

Game Analytics is an important research topic in digital entertainment. Data log is usually the key to understand players' behavior in a game. However, alpha and beta builds may need a special attention to player focus and immersion. In this paper, we propose the use of player's focus detection, through the classification of pictures. Results show that pictures can be used as a new source of data for Game Analytics, feeding developers with a better understanding of players enjoyment while in testing phases.

KEYWORDS

Game Analytics, Fuzzy, FLA, Game Flow, Player Focus

1 INTRODUCTION

Game developers must be able to understand players' profile and in-game enjoyment [1]. This kind of information can provide a comprehension of games problems. Usually, players' insights are obtained by character's behavior [2].

Camera stream video can be used to detect screen focus [3], and this information can be used along with in-game logs to enhance data analytics. We expect that this model helps developers to better understand players reactions during the testing phase, allowing assessments of players immersion, and game design problems.

2 MODEL PROPOSAL

This paper proposes a model for detecting player focus through computer vision. For this, we developed an API that receives images taken during gameplay and creates a heatmap showing players attention.

To discover the face position, facial point annotation techniques were used. They search for the landmarks that

characterize a face [4]. The result was then classified by a fuzzy model.

2.1 Points of Interest

Most facial points annotation techniques are performed using pre-trained activation models to find point from a given Region of Interest (ROI) with the face to be analyzed [5].

This model was trained with heterogeneous databases and can generate good results. However, this technique needs a pre-processed rectangle containing the face. To do it, a Cascade Classification was used.

2.2 Haar Cascade

Haar Cascade is an Artificial Intelligence model that is known for being fast and efficient [6]. There are lots of implementations, including the OpenCV library, which is the one used in this work.

Thus, the automatic screen focus detection used by this model is based on using annotating points. Since it is not necessary to know the exact point a player is looking at, but only if he is watching the game screen or not, this method should be the most efficient. It is expected that it will be able to tell whether the user is looking at the computer screen or not.

2.3 Implementation

For finding faces in an image, we used the "haarcascade_frontalface_alt" model. This model returns an array of faces, when the array had a length greater than 1, the more prominent (larger) face was selected as the main face. With the face ROI, we could use it as input to the next algorithm, which consists of detecting 68 landmark points on it. For this, we used the points of interest model FacemarkLBF [7].

To detect direction, a simple calculation is used, based on a representation of those points. For this, the number of points was reduced to 6: the chin, the nose, the rightmost point of the right eye, the leftmost point of the left eye, the rightmost point of the mouth, and the leftmost point of the mouth. Detecting eye

direction (pupil) was not possible, because of the image resolution, and lightning problems when using glasses.

With these six points, we can identify the direction the person is looking, based on the rotation rate described in (1). Where: x_n represents the nose, x_{lm} , the left end of the mouth, and x_{rm} the right end of the mouth.

$$RotationRate_{f_{x,y}} = \frac{|x_n - x_{lm}|}{|x_n - x_{lm}| + |x_n - x_{rm}|} \quad (1)$$

Rotation Rate (1) calculates a value between 0 and 1. Since this value is hard to understand, we created a fuzzy model that better describe this variable, where all three classes (right, center, and left) are distributed in a normal curve. To calculate the membership, we used scikit-fuzzy library in Python. An inclination rate was used to calculate it vertically. The rate is described by (2), where y_n is the y-position of the nose, y_{le} the left eye, y_{re} the right eye, and y_c the chin.

$$InclinationRate_{f_{x,y}} = \frac{(|y_n - y_{le}| + |y_n - y_{re}|) / 2}{|y_n - y_c|} \quad (2)$$

The result is a value between 0 and 2, modeled in 3 fuzzy classes: top, middle, and bottom. That way, it is possible to predict whether the person has his face turned towards the computer screen or not. Since we made this processing as an API, sending a message we get a response with rotation and inclination rates, and a textual direction label.

This response can be used as a single information, or with a historical motion, to understanding movement or finding the camera position.

3 TESTS

To test the capacity of the method, it was tested with 144 images captured on Twitch (a video live streaming service, focused on video game players). Twitch was used because is similar to the environment expected during gameplays, with a person using the computer while being filmed by a webcam. Also, Twitch's register and image rights policies allow the sharing and reproduction of clips. These rules warrant the creation of image and video databases without complexities related to their capture and copyrights.

Images were selected at random, in streams that were being reproduced at the time of collection. Regarding the rotation of the faces (left, center or right), from the 144 manually cataloged images, 18 had problems with face detection that did not allow the application to run.

This happens when the ROI detection algorithm cannot find any faces on the screen. All those that fit this situation were due to lack of lighting, or because the face is too rotated to the camera (in the second case they should be classified as "not looking at the camera"). These results are shown in Figure 1.

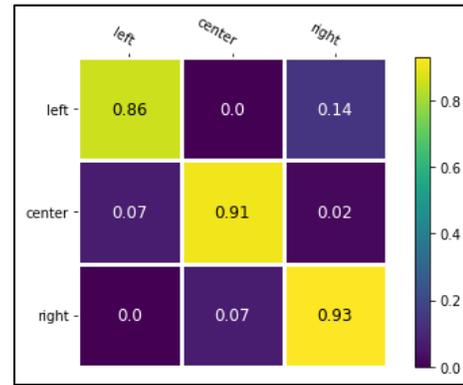


Figure 1: Confusion Matrix for rotation rate.

Among the processed images, 12 had some failure in position prediction. Thus, 114 were correct (90% of the 126 processed). Most errors are due to problems in detecting the rectangle of interest in the face, and the interest points. For the inclination, the correctness was 82.5%. Images that had errors were also the ones with problems in detecting points.

4 CONCLUSIONS

By using players' images to detect focus, we can have a better comprehension of the game, and the moments that are interesting and immersive. This type of analysis is usually done manually during gameplay tests, and by automating them it is possible to apply these tests at scale.

Problems detected were mostly in landmark annotations. As future works, we are going to use alternatives for this job, probably other models and techniques like Convolutional Neural Networks may provide us with face meshes better suited for this purpose.

We are already using this model with in-game logs to create an automated Game Analytics model. So, developers can automate the whole game test information capture process, by using early access builds and collecting information from early players.

REFERENCES

- [1] D. K. Jain, P. Shamsolmoali, and P. Sehdev, "Extended deep neural network for facial emotion recognition," *Pattern Recognit. Lett.*, vol. 120, pp. 69–74, 2019, doi: 10.1016/j.patrec.2019.01.008.
- [2] A. Steffens Henrique, R. Martins Brasil Soares, R. Luis Scaranto Dazzi, and R. Lyra, "Genetic Algorithm in Survival Shooter Games NPCs," in *Computer on the Beach*, 2020, pp. 413–418, doi: 10.14210/cotb.v11n1.p413-418.
- [3] S. M. Munn and J. B. Pelz, "3D point-of-regard, position and head orientation from a portable monocular video-based eye tracker," in *Eye Tracking Research and Applications Symposium (ETRA)*, 2008, pp. 181–188, doi: 10.1145/1344471.1344517.
- [4] B. Egger et al., "3D Morphable Face Models-Past, Present, and Future," *ACM Trans. Graph.*, vol. 39, no. 5, p. 157, Sep. 2020, doi: 10.1145/3395208.
- [5] G. Tzimiropoulos, J. Alabort-I-Medina, S. P. Zafeiriou, and M. Pantic, "Active orientation models for face alignment in-the-wild," *IEEE Trans. Inf. Forensics Secur.*, vol. 9, no. 12, pp. 2024–2034, 2014, doi: 10.1109/TIFS.2014.2361018.
- [6] S. Soo, "Object detection using Haar-cascade Classifier." 2014.

- [7] J. Yan, W. Zheng, Z. Cui, C. Tang, T. Zhang, and Y. Zong, "Multi-cue fusion for emotion recognition in the wild," *Neurocomputing*, vol. 309, pp. 27–35, 2018, doi: 10.1016/j.neucom.2018.03.068.