# Player focus tracking based on computer vision for Game Analytics improvements

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

Rudimar Luis Scaranto Dazzi
*Laboratório de Inteligência Aplicada*
*Univali*
Itajaí, Brasil
rudimar@univali.br

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

Anita Maria da Rocha Fernandes
*Laboratório de Inteligência Aplicada*
*Univali*
Florianópolis, Brasil
anita.fernandes@univali.br

Rodrigo Lyra
*Laboratório de Inteligência Aplicada*
*Univali*
Itajaí, Brasil
rlyra@univali.br

***Abstract*— Game Analytics became an important research topic in the digital entertainment field, due to the increase in machine learning and artificial intelligence frameworks available in the game industry. While most game analytics approaches are based on naive data log collection, there is much possible information that can be included in the inference process, enhancing, and giving more precision to the prediction and classification process. In this paper, we propose and present the usage of real-time player attention and focus data through the analysis and classification of camera stream video. We intend to include the emotional state of the player into regular game analytics data, feeding game designers, and producers with more accurate statements about the player emotions and immersion.**

***Keywords—Image Processing, GameAnalytics, Fuzzy, Deep Learning, Emotion Recognition.***

## I. INTRODUCTION

With big improvements in data achievement and machine learning processes, it is becoming a trend to better understand players' profiles and characteristics [1]. In games, the understanding of human emotion can be obtained, based on character's behavior [2], or player's information [3].

In-game human behavior modeling techniques are mainly used for modifying game difficulty or evaluating players' learning and game's quality. There are many variables that can be used for game analytics processes. We believe that emotional feedback can be one of them, enhancing the understanding of gameplay.

This paper aims to present a model for analyzing players' facial images during gameplay. This model can detect players' emotions as well as screen focus. As limitations, the model considers that the camera will be positioned at the top center of the screen (like notebook cameras).

We expect that this model helps developers to better understand players during the testing phase, allowing assessments of players satisfaction, based on emotional engagement and feedback.

## II. RELATED WORKS

Game Flow [4] is one of the main models for evaluating players' enjoyment. It is made of eight elements, to model enjoyment as a number. It is very efficient and widely used in game development companies, although it is a manual process and requires experts to make the evaluation based on players' experience.

Trying to make this kind of test more automated, models use game logs to understand players' enjoyment without the need for manual analysis [5]. They also use physiological responses to understand players' emotions while playing [6].

With deep learning and image processing, many kinds of research work in recognizing human emotion based on image or video records. Pandey [7] shows that it's possible to automate facial emotion recognition in images, by using Convolutional Neural Networks (CNNs).

Hu [8] uses emotion history to better predict facial emotion, by landmarking these images before feeding the network. Using this kind of method, developers can get players' image sequences and convert it to knowledge.

In general, game analysis models, use human observation, game log, playing reports, or physiological responses. We believe that using images, we can not only improve game analytics data collection but also find new kinds of information that were not possible without it.

## III. MODEL PROPOSAL

In this paper, we propose a model for collecting real-time player attention and focus data through image processing. These images were taken during gameplay and sent to an API that processes players' emotions.

The model has two main parts, one responsible for screen focus, and another for emotion recognition. We propose to capture emotional states based on faces aspects transitions. A transition diagram was created to weigh players' expression, prioritizing specific changes. These transitions are represented in Table 1, where lines represent the source, and columns the next vertex.

TABLE I. EMOTIONS ADJACENCY MATRIX

| from/to | calm | happy | sad | angry |
|---|---|---|---|---|
| **calm** | 1 | 5 | -5 | -3 |
| **happy** | -1 | 3 | -7 | -2 |
| **sad** | 3 | 10 | -5 | -10 |
| **angry** | 1 | 10 | -3 | -1 |

This adjacency table provides the weights, so the system can track emotions changes and calculate an enjoyment score. These weights help the system to better model human emotions while playing, mostly by given higher values to more relevant transitions, that can make the player more or less interested in the game.

Weights were defined so the greater the final score, the greater is the probability of the player to replay the game. That way, transitions that are considered bad, have negative weights, while good ones have positive. These weights can be changed depending on the game scope. The score can be calculated for the whole game (for each player), or for a specific level (for each level played by each player).

For player focus, we used a fuzzy classification model and its values to create a heatmap that let us understand players' focus through gameplay.

## IV. SCREEN FOCUS

Screen focus detection is usually performed by Eye-Tracking devices [9]. However, some researchers seek to do this by processing images captured by the user's webcam, but the required processing for this is complex [10]. To discover the face position, facial point annotation techniques are used. They search for the milestones that characterize a face [11].

### A. *Finding the face*

Currently, most facial points annotation techniques are performed using pre-trained activation models. They can find points of interest, given a rectangle of interest (ROI) with the face to be analyzed [12]. These points represent the main features in a face, and can be used both in emotion recognition, and screen focus [13]. This technique needs a pre-processed rectangle that covers the face. To do it, a Cascade Classification was used. It is expected that its information will be able to tell whether the user is looking at the computer screen or not.

### B. *Implementation*

The screen focus analyst's role is to detect whether the player's focus of attention is on the screen or not. For this, an image processing-based approach was used. The service was made available as a Python API.

First, it is necessary to find the faces in the image. For this, we used the model "haarcascade_frontalface_alt", and the more prominent face was selected as the main face. Having the face ROI, it is possible to use the next algorithm, which consists of detecting 68 interest points on it. For this, we used the points of interest model LBF.

To detect the direction the person is looking at, a simple calculation is used, based on a simplified representation of the points. Initially, the number of points on the face is reduced to 6, which represents: 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.

Knowing these six points, it is possible to identify the direction in which the person is looking, based on the rotation rate, which is 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.

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

Equation (1) calculates the rate of face rotation. This number is a value between 0 and 1, which represents how horizontally rotated the face is. Since this change is not absolute, fuzzy modeling was performed for this variable. The relevance of this modeling classes can be interpreted according to Fig. 1.

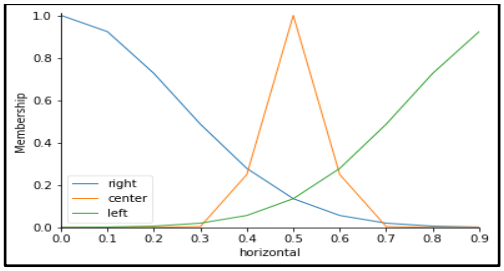


Fig. 1. Fuzzy classes to face rotation

Where all three classes (right, center, and left) are distributed in a normal curve, with 0, 0.5, and 1 as means and deviations of 0.25, 0.06 and 0.25, respectively. The membership in each class is calculated for the Rotation Rate using the scikit-fuzzy library.

To evaluate the face vertically, an inclination rate was used. 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.

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

As a result, a value between 0 and 2 is obtained, where fuzzy modeling of 3 classes: top, middle, and bottom, is applied. They are also Gaussian distributions, with an average of 0, 0.66 and 1.33, and deviations of 0.3, 0.2, and 0.25, respectively. That way, it is possible to predict whether the person has his face turned towards the computer screen or not.

## V. FACIAL EXPRESSION EMOTION RECOGNITION

Facial expressions are one of the most important signs used by humans to demonstrate their intentions and their emotional state [14]. With the increase in computing power, deep learning techniques started to be used for this kind of task [15]. With these techniques, software can achieve much better results in human understanding [16].

The main emotion recognition services are Google Cloud Vision, Microsoft Azure Computer Vision, and Amazon Rekognition. Since all these services are paid and with closed technologies, it is not possible to accurately discover the AI techniques used. Also, each emotion recognition service in images has a different interface, analyzing different types of emotions and with different metrics, as shown in Table 2.

TABLE II. EMOTIONS RECOGNITION APIS METRIC

| Service | Emotions | Metrics |
|---|---|---|
| **Google** | 4 | 1 - 5 |
| **Azure** | 8 | 0,0 – 1,0 |
| **Rekognition** | 7 | 0,0 – 99,99 |

Google's API presents values corresponding to emotions: happy, angry, sad and surprised. Microsoft's, happy, sad, angry, contempt, disgusted, surprised, calm and fear. And Amazon's, happy, sad, angry, confused, disgusted, surprised, calm.

Given the interface difference for each platform, their results require different interpretation strategies. For better comprehension, a test with an image was performed (Fig. 2). And with these results, it was possible to better understand the modeling of each service.

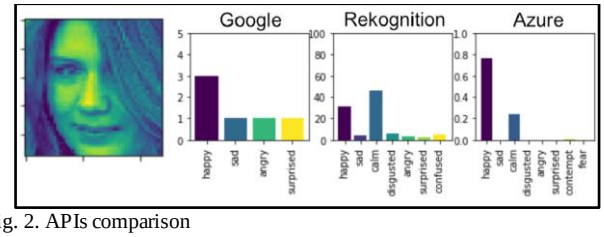


Fig. 2. APIs comparison

Fig. 2 presents a picture, followed by the results of the Google, Rekognition, and Azure APIs. The metrics used by Google considers the image kind of happy. Both Azure and Rekognition have percentages, but the results are different. Microsoft's model shows an emotion that is 75% happy. Amazon, however, classifies the image primarily as calm, 46%, and then happy, 30%.

Seeking for more conclusive and reliable results on these tools, tests with a larger image dataset were done. The test was performed using images from the FER2013 database. To get a more accurate comparison between the APIs, only images labeled as happy, angry, sad and calm were used. Each experiment was categorized as: correct, when the expression cataloged was the same as that of the database, error when the expression had a different expression and fail, in cases where the algorithm was unable to recognize a face in the image. Fig. 3 demonstrates the distribution of hits, errors, and fails for each API.

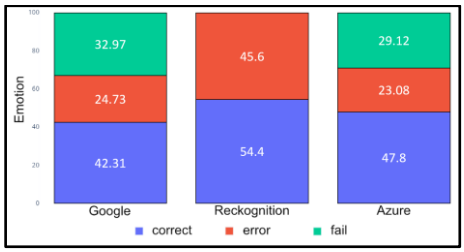


Fig. 3. Emotion Recognition Results

Azure and Vision APIs achieved similar results when exposed to this data. Failing to recognize faces in about 29% of cases. Azure, however, is more faithful to the expected results, with an average of 47.8% hits.

The Rekognition API was able to detect faces in all situations and had an average hit of 54.4%. This adjustment, considering only the images in which faces were detected, is inferior to that of the other APIs, which achieved an average of 65% correctness when detecting the faces. However, considering all cases (including non-face recognition) it is a more satisfactory value. To better understand emotion predictions, APIs results were cataloged as a confusion matrix presented in Fig. 4.

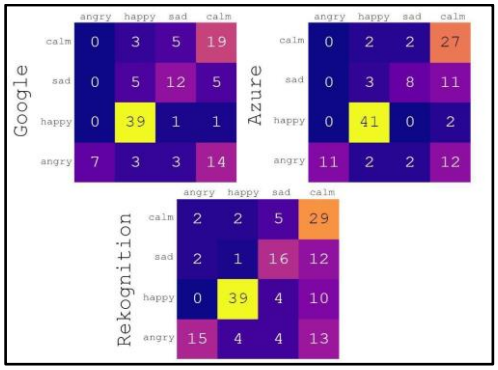


Fig. 4. Emotion Recognition APIs Confusion Matrix

Tests show greater assertiveness in detecting happiness, although showing a certain margin of false positives. Anger was the most reliable, with a few false positives, in contrast to neutrality (calm), which had the highest false positives rate. This is because not so prominent emotions can be labeled as calm.

These tests show that Emotions Recognitions APIs can be used, but their predictions are still not enough. For the in-game tests, we used both predicted emotions and hand-labeled ones.

## VI. TESTS AND RESULTS

The model was tested in two games developed in Unity. Both were manually rated (using GameFlow) before the test and had different scores. For testing purposes, we have decided to use two types of games: the first is a game with known gameplay problems and the second is a well-finished game, with better players´ enjoyment. The main reason for that is to look for indications that the model cannot only score a good game better than a bad one. But also, to point out the worse levels (according to players' enjoyment).

Game 1 is an isometric puzzle game, which is still in early development. It has some major bugs, and the gameplay is not complete yet. As some levels were better than others, we expected to be able to see this difference in the results.

Game 2 is a 3D third-person puzzle, made by Unity. Although it is a tech demo, it has two levels and can be played to the end. It was considered a good game and having only two levels, no big changes between them were expected.

Both were injected with a component that takes pictures, whenever the player does something that changes the game state (like activating a button or finishing a level). After collected, those pictures were processed so the main emotion and screen focus could be predicted.

The screen focus was plotted as a heatmap, which shows the player's facial position according to the camera. This plot can be seen in Fig. 5, where Player A and B are compared.

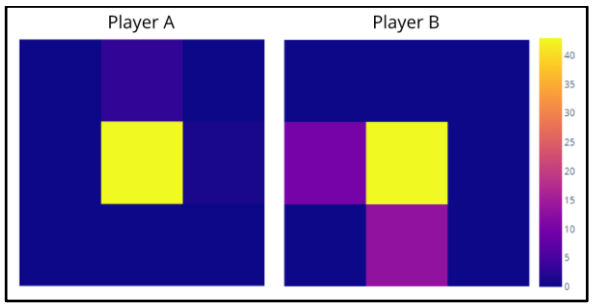


Fig. 5. Players focus heatmap

When comparing, we can see that player A was a lot more focused on the screen, as his heatmap shows, he rarely moved his face enough to be considered by the model. On the other hand, player B moved a lot his focus to the left and down.

Using this technique by level, we can see which levels are more likely to let the player unfocused. For example, we can see in Fig. 6 that players often lose their focus on-screen in level 1, but not in level 6.

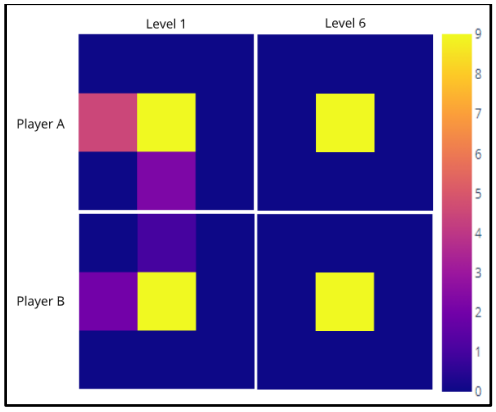


Fig. 6. Focus Heatmap by level

Focus is still not enough for understanding players' enjoyment, that is why the model has an emotion-based score. These scores can be calculated for the entire game, like Game 1 that has a score of -2.052, and Game 2 1.059. To do so, we calculated the transitions score, and divided by the number of entries.

We can calculate it by level, so we can better understand where the main problems of our game are, as shown in Fig. 7, where the x-axis is the game level, and y the normalized emotion score, for each room in Game 2 (since it has only two levels, the room approach was used to collect more specific and relevant information).

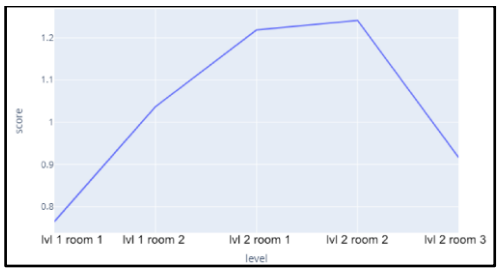


Fig. 7. Game 2 Emotion score by room

This way, we can see that, at least, level 1-1 and level 2-3 need some rework. And people are usually less happy while playing them. Lastly, we can calculate this score by player and by level, so we have a better understanding not only about our game but also about each player journey and differences in enjoyment by group.

## VII. CONCLUSIONS

The usage of information from players' images makes it possible for the developer to extract new information, which allows him a better understanding of its users.

Results show that the pre-trained models for detecting faces and emotions can already predict useful emotions, that provides developers with a deeper understanding of their users. Although new emotions like boredom may help improve these results. We believe that training new facial emotion recognition models, especially with images of players while in-game, would provide better results.

The model described in this paper allows a better understanding of the players' focus and immersion. This type of analysis is usually done manually during gameplay tests, and by automating them it is possible to apply these tests at scale.

We are already using these two models with in-game logs to create an automated model, able to provide evaluations that are similar to the ones GameFlow can give. So, developers can automate the whole game test information capture process, by using early access builds and collecting information from early players.

## ACKNOWLEDGMENTS

This work was supported by the Coordination for the Improvement of Higher Education Personnel - Brazil (CAPES) - Financing Code 001

## REFERENCES

[1] Deepak Kumar Jain, Pourya Shamsolmoali, and Paramjit Sehdev, "Extended deep neural network for facial emotion recognition," *Pattern Recognition Letters*, 2019.

[2] Alisson Steffens Henrique, Ricardo Martins Brasil Soares, Rudimar Luís Scaranto Dazzi, and Rodrigo Lyra, "Genetic Algorithm in Survival Shooter Games NPCs," *XI Computer on the Beach*, 2020.

[3] Aurélien Appriou, Andrzej Cichocki, and Fabien Lotte, "Modern machine learning algorithms to classify cognitive and affective states from electroencephalography signals," *IEEE Systems, Man and Cybernetics Magazine*, 2020.

[4] Penelope Sweetser and Peta Wyeth, "GameFlow: a model for evaluating player enjoyment in games," *Computers in Entertainment (CIE)*, vol. 3, pp. 3-3, 2005.

[5] Sabine Trepte and Leonard Reinecke, "The Pleasures of Success: Game-Related Efficacy Experiences as a Mediator Between Player Performance and Game Enjoyment," *Cyberpsychology, Behavior, and Social Networking*, 2011.

[6] Simone Tognetti, Maurizio Garbarino, Andrea Bonarini, and Matteo Matteucci, "Modeling enjoyment preference from physiological responses in a car racing game," *Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games*, 2010.

[7] Ram Krishna Pandey, Souvik Karmakar, AG Ramakrishnan, and Nabagata Saha, "Improving facial emotion recognition systems using gradient and laplacian images," *arXiv preprint arXiv:1902.05411*, 2019.

[8] Min Hu, Haowen Wang, Xiaohua Wang, Juan Yang, and Ronggui Wang, "Video facial emotion recognition based on local enhanced motion history image and CNN-CTSLSTM networks," *Journal of Visual Communication and Image Representation*, 2019.

[9] Emila Cubero Dudinskaya, Simona Naspetti, and Raffaele Zanoli, "Using eye-tracking as an aid to design on-screen choice experiments," *Journal of Choice Modelling*, 2020.

[10] Jorge Candido, "Detecção e rastreio de faces utilizando redes Bayesianas," Universidade Presbiteriana Mackenzie, 2007.

[11] Bernhard Egger et al., "3D Morphable Face Models—Past, Present, and Future," *ACM Transactions on Graphics (TOG)*, 2020.

[12] Georgios Tzimiropoulos, Joan Alabort-i-Medina, Stefanos P Zafeiriou, and Maja Pantic, "Active orientation models for face alignment in-the-wild," *IEEE transactions on information forensics and security*, 2010.

[13] Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic, "A semi-automatic methodology for facial landmark annotation," *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2013.

[14] Gabrielle Simcock et al., "Associations between facial emotion recognition and mental health in early adolescence," *International journal of environmental research and public health*, 2020.

[15] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv*, 2014.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.