

Modelling Virtual Camera Behaviour Through Player Gaze

Andrea Picardi
Center For Computer Games Research
IT University Of Copenhagen
Rued Langgaards Vej 7
2300 Copenhagen
anpi@itu.dk

Paolo Burelli
Center For Computer Games Research
IT University Of Copenhagen
Rued Langgaards Vej 7
2300 Copenhagen
pabu@itu.dk

Georgios N. Yannakakis
Center For Computer Games Research
IT University Of Copenhagen
Rued Langgaards Vej 7
2300 Copenhagen
yannakakis@itu.dk

ABSTRACT

In a three-dimensional virtual environment, aspects such as narrative and interaction largely depend on the placement and animation of the virtual camera. Therefore, virtual camera control plays a critical role in player experience and, thereby, in the overall quality of a computer game. Both game industry and game AI research focus on the development of increasingly sophisticated systems to automate the control of the virtual camera integrating artificial intelligence algorithms within physical simulations. However, in both industry and academia little research has been carried out on the relationship between virtual camera, game-play and player behaviour. We run a game user experiment to shed some light on this relationship and identify relevant differences between camera behaviours through different game sessions, playing behaviours and player gaze patterns. Results show that users can be efficiently profiled in dissimilar clusters according to camera control as part of their game-play behaviour.

Categories and Subject Descriptors

I.2.1 [Artificial Intelligence]: Applications and Expert Systems—*Games*;

H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—*Evaluation/Methodology*

1. INTRODUCTION

The virtual camera plays a vital role in 3D computer games since it directly affects player experience and game enjoyability [15]. A virtual camera serves as the player's point-of-view through which she perceives the game world and gets feedback on her actions. Camera placement and animation in games is usually directly controlled by the player or statically predefined by designers. However, direct control of the camera by the player increases the complexity of

the interaction and reduces the designer's control over game storytelling. On the other hand, statically placed cameras release the player from the burden of controlling the point of view, but they cannot guarantee a correct visualization of all possible player's actions. Moreover, in multi-player games or in games where the content is procedurally generated, the designer has potentially no information to define a-priori the camera positions and movements. Automatic camera control aims to define an abstraction layer that permits to control the camera using high-level and environment-independent requirements, such as the visibility of a particular object or the size of that object on the screen. Given these requirements and the current game state, the camera controller should dynamically and efficiently infer the ideal camera position and motion.

The process of finding the virtual camera position and orientation that best fit a set of requirements has been widely investigated [11]. On the contrary, the requirements themselves have received little attention despite their impact on player experience [14]. Different camera settings, e.g. camera position or number of subjects on the screen, affect the information the player perceives from the game and influence the player's emotional state. Moreover, the camera has a significant impact on interaction as the vast majority of modern computer games base their game-play on visual actions. It would be impossible, for instance, to aim at a target which is not visible in the frame, or too easy to look for an object which is always in the center of the screen.

Virtual camera parameters are commonly hand-crafted by game designers and do not consider player preferences. Including the player in the definition of these parameters requires the construction of a model of the relationship between camera motion and player experience [14]. In this paper we investigate player preferences on virtual camera placement and animation and analyse the relationship between camera behaviour, player behaviour and game-play. In order to understand this relationship, data from player gaze and the virtual camera motion is collected through a game experiment and used to identify and describe the players' camera behaviours. Our hypothesis is that the gaze motion behaviour, which is directly linked to the manual camera control behaviour, depends on the player and the type of her game-play actions and preferences. Therefore, we expect that different players will focus on different object types in the same game situation and that the same player will observe different types of object in different game situ-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

FDG '11, June 29-July 1, Bordeaux, France.

Copyright 2011 ACM 978-1-4503-0804-5/11/06 ...\$10.00.

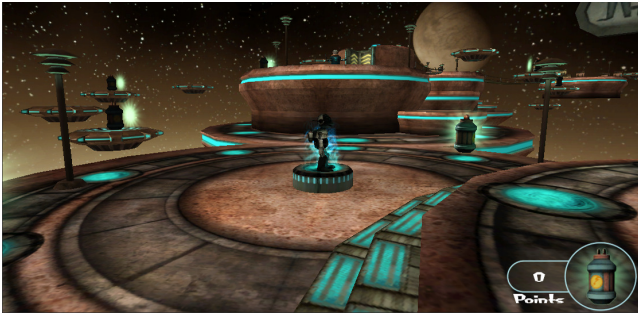


Figure 1: A screen-shot of the game used in the experiment.

ations.

The test-bed game used for all experiments in this paper is a three-dimensional platformer (see Fig. 1) featuring all the stereotypical aspects of the genre’s mechanics. The player is requested to control the avatar and the camera through the game while jumping on platforms, fighting with enemies and collecting objects. Including gaze information in the experiment allows a finer analysis of the player’s visual behaviour permitting, not only to understand what objects are visualised by the player, but also which ones are actually observed. Information on player’s visual focus also allows to filter exactly which object is relevant for the player among the ones visualised by the player through her control of the virtual camera. Results obtained from the study presented in this paper confirm the hypothesis above as relevant differences in the camera behaviour were found among groups of players and among game situations. A cluster analysis of the gaze data collected reveals the existence of different virtual camera motion patterns among the different areas of the game. Moreover, a statistical analysis of the game-play features between the identified clusters demonstrates the influence of certain game-play aspects on camera behaviour.

The paper provides an overview of the related work on camera control and gaze interaction in games (Section 2) and continues with a description of the test-bed game employed in the experiment (Section 3) and a description of the experimental methodology followed (Section 4). Section 5 lists the statistical features extracted from the collected data and Section 6 provides a discussion of the results obtained. The paper’s main findings are summarised in Section 7.

2. RELATED WORK

Early studies on camera control focused on the definition of the camera properties and investigated the mapping between input devices and camera movement [17, 6]. The main research focus in the field rapidly shifted towards automatic camera control since direct control of the camera has shown to be problematic for the user [12].

Several different techniques have been proposed for automatic camera control, based on a variety of mathematical models; however, the vast majority of the approaches model the problem as a constraint satisfaction or optimisation problem [11]. These approaches allow the designer to define a set of requirements on the frames that the camera should produce and on the camera motion. Depending on the approach, the controller positions and animates one or more virtual cameras in the attempt to generate a shot or a sequence of shots that satisfy the predefined requirements.

Requirements for the camera include constraints on camera motion (such as speed limit), constraints on camera position (such as maximum height), or constraints on the rendered frames. The last type of camera requirements, introduced by Bares et al. [3], define required properties on the frames rendered using the camera such as subject inclusion or subject position within the frame.

2.1 Camera profiles

A large volume of research studies on automatic camera control is dedicated to the analysis of robust and time-efficient techniques to place and move the camera to satisfy a set of given requirements. The definition of the requirements for the camera motion and placement is often delegated to an hypothetical designer, but questions such as “which subject should be in the frame?”, “where should it be in the frame?”, or “when should it appear?” have triggered a large number of research studies within the virtual cinematography and game AI research fields. These sets of requirements, usually referred as *camera profiles* [7], define the set of constraints and the objective function that needs to be optimised by the automatic camera control system.

Christianson et al. [9] introduced a language that permits the definition of shot sequences (idioms) with the desired timings and subjects. Other researchers extended Christianson’s work by connecting shot plans with camera constraints [1], or by introducing advanced planning techniques to support interactive storytelling [8, 13]. While the aforementioned approaches address primarily the issues related to the manual design of camera behaviours for dynamic and interactive environments, other researchers have investigated a different direction which does not require the contribution of a designer. For instance, Bares’s and Lester’s work on cinematographic user models [2] exploits user preferences to define personalised camera profiles. One year later, Bares et al. [4] developed a framework based on a model of intention that automatically chooses a camera profile depending on the user’s task.

Yannakakis et al. [18] studied the impact of camera viewpoints on player experience and built a model to predict this impact. That research study demonstrates the existence of a relationship between player emotions, physiological signals and camera parameters. Therefore, in the light of these results, it appears important to further investigate the relationship between camera and player experience to automate the generation and selection of the virtual camera parameters. In this paper we attempt to incorporate alternative player input modalities (i.e. gaze) as information for camera profiling.

2.2 Gaze interaction in games

Eye movements can be recognised and categorised according to speed, duration and direction [19]. In this paper, we focus on *fixations*, *saccades* and *smooth pursuits*. A fixation is an eye movement that occurs when a subject focuses at a static element on the screen; a saccade occurs when a subject is rapidly switching her attention from one point to another and a smooth pursuit is a movement that takes place when a subject is looking at a dynamic scene and she is following a moving object.

Sundstedt et al. [16] conducted an experimental study to analyse players’ gaze behaviour during a maze puzzle solving game. The results of their experiment show that gaze

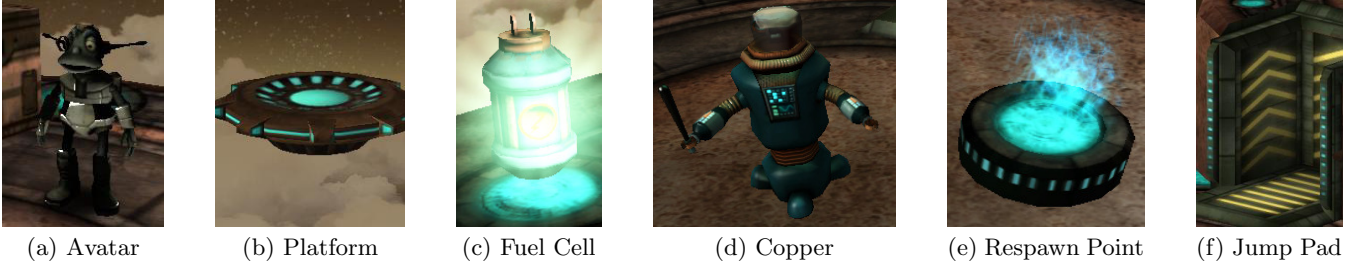


Figure 2: Main components of the test-bed game used.

movements, such as fixations, are mainly influenced by the game task. They conclude that the direct use of eye tracking during the design phase of a game can be extremely valuable to understand where players focus their attention, in relation to the goal of the game. Bernhard et al. [5] performed a similar experiment using a three-dimensional first-person shooting game in which the objects observed by the players were analysed to infer the player’s level of attention. We are inspired by the experiment of Bernhard et al. [5]; unlike that study however, we analyse the player’s gaze patterns to model the player’s camera movements, and moreover, investigate the relationship between camera behaviour, game-play and player-behaviour.

3. THE GAME

A three-dimensional platform test-bed game has been designed for the purpose of this study¹. The game features a robot-like avatar (see Fig. 2a) in a futuristic 3D environment made of floating platforms (see Fig. 2b). The platforms can be flat, or be composed by multiple smaller platforms with different heights clustered together. Each platform can be connected to another platform through a bridge or be disconnected, in which case the avatar is required to jump to move from one platform to the other.

There are four main elements/objects that appear in this game: *fuel cells*, *coppers*, *re-spawn points* and *jump pads*. Fuel cells (see Fig. 2c) are collectable items increasing the score of the player. Coppers (see Fig. 2d) are animated robots which chase the avatar and hit him until he falls from a platform. Coppers are normally static and get activated when the avatar enters the platform they are placed on. The player can kill a copper by moving the avatar close to it and hitting it three times; killing a copper increases the player’s score. Re-spawn points (see Fig. 2e) are small stands placed on some platforms. When the avatar touches a re-spawn point this becomes activated; each time the avatar falls from a platform it reappears on the last activated re-spawn point. Finally, the jump pads (see Fig. 2f) are black and yellow areas on the platforms which allow the player to perform a long jump.

The aim of the player is to reach the end of the level and achieve the highest possible score. The score depends on the time passed to reach the end of the level, the number of collected items, the number of times the player avatar falls from a platform and the number of damage inflicted to the enemies. The total score, S , is calculated as $S =$

$5n_{fc} + 5n_p - 10n_{fa} + 15n_k + 150\frac{L_e}{L_a}$ where n_{fc} is the number of fuel cells collected by the player, n_p is the number of times the player hits a copper, n_{fa} is the number of times the avatar falls from a platform, n_k is the number coppers killed, L_e is the expected length of the level in seconds and L_a is the actual time spent by the player to complete the level. The score is designed to foster for different playing strategies. Since the contributions of the different actions have a comparable impact to the final score it is left to the player to decide which action is more convenient to perform according to her playing style.

The game consists of four levels: an initial tutorial and three main levels. Each level starts with the avatar standing on the initial re-spawn point, as shown in Fig. 1, and ends when the avatar reaches the last platform. The tutorial level includes a walk-through of the game controls and an overview of the contribution of the game actions to the score. Moreover, during the tutorial, the player is guided through all the challenges she will encounter in the three successive levels. To eliminate any potential order effects the order of the subsequent three levels is randomised and different for each player. The main levels A, B and C, depicted in Fig. 4, have respectively an estimated gameplay length of 1 minutes and 36 seconds, 3 minutes and 12 seconds, and 2 minutes and 50 seconds. The expected play durations have been determined measuring the competition time of an expert platform player.

3.1 Controls

The player directly controls the avatar, as well as the camera position and the camera orientation. The avatar is controlled using the keyboard keys while the camera is controlled via the mouse as described in Table 1. The avatar movements are defined in a camera-relative space, therefore the avatar will move left, right, forward and backward relatively to the player’s point of view. The camera follows the character as an orbital object at a fixed distance; the player

Button	Action
W, A, S, D keys	Move the avatar (Forward, Backward, Left, Right)
Mouse movement	Direct orbital camera control
Mouse Wheel	Camera zoom
Left mouse click	Punch
Right mouse click	Jump
Centre mouse click	Run

Table 1: Player’s controls via the keyboard and mouse.

¹The game is based on Lerpz, a tutorial game by Unity Technologies - <http://www.unity3d.com>

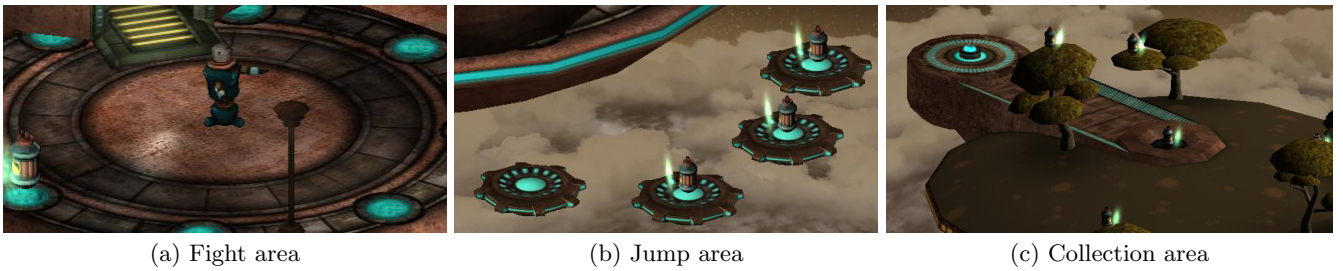


Figure 3: The three different area types met in the test-bed game.

is able to change the distance using the mouse wheel and rotate the camera around the avatar by moving the mouse. Moving the mouse forward will increase the height of the camera while moving it backward will decrease it. Moving the mouse sidewise will make the camera rotate around the avatar according to the mouse direction.

3.2 Game-play Areas

The three game levels are synthesised by a series of sub-levels we name *areas* which are classified as *jump*, *fight* or *collection* areas. The areas are categorised according to the game-play experience they offer and the type of challenge they pose to the player. In case an area offers more than one challenge type, the category is defined by the most threatening challenge. The challenges are sorted in decreasing level of threat as follows: fight, jump and collect.

Figure 3a shows a fight area where the main threat is given by the opponent copper at the center of the platform. The jump area depicted in Fig. 3b is composed by several small floating platforms; the player needs to make the avatar jump across all the platforms to complete the area. Figure 3c shows an area where the main task of the player is to collect the fuel cells placed around the platform.

4. EXPERIMENTAL METHODOLOGY

As mentioned earlier, our experimental hypothesis is that the way a player controls the virtual camera in a three dimensional computer game is directly linked to her gaze behaviour and to the actions she is performing in the game. We represent the virtual camera behaviour as the amount of time the player spends framing and observing different objects in the game environment while playing the game. Specifically, in this experiment, the camera behaviour model is based on the amount of time the player spends watching fuel cells, enemies, platforms, jumps pads and re-spawn points. These times are measured in the experiment by combining the player’s gaze position on the screen with the virtual camera orientation.

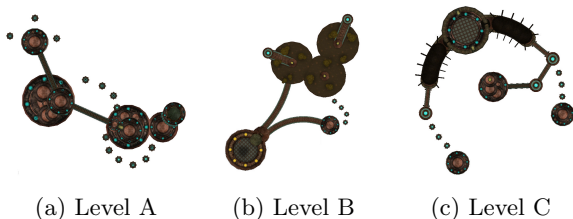


Figure 4: The three levels of the test-bed game.

4.1 Setup

The experiment is set in a room without windows. The room contains a desk, a chair and two computers: the first computer runs the computer game while the second one collects the data from the game and the player’s gaze and synchronises the two streams into one log. The first computer is connected to a 17” screen, a keyboard and a mouse; the second computer is headless and is only connected to the gaze tracker and, through a network cable to the first computer (see Fig. 5). Running the gaze tracking process and the game on different machines guarantees the necessary resources for both and minimizes the risk of data loss due to heavy computation.

We used the *Eyefollower*² gaze tracker for the purpose of this experiment. The Eyefollower is a device able to locate the player’s gaze position on a computer screen through a combination of three cameras. The first camera, placed on top of the screen, tracks the head position while two other motorised infra-red cameras, placed at the bottom of the screen, follow the player’s eyes and track the point of regard (Fig. 5). The gaze tracker samples the player’s gaze at a 120 Hz frequency (60 Hz per eye). The declared accuracy of this system is 0.45 degree in optimal conditions and the visual acuity is about 1 degree. The Eyefollower necessitates an environment with fluorescent lighting and no daylight or incandescent lights as ambient infra-red light or incandescent lights would interfere with the infra-red light emitted by the gaze tracker.

4.2 Protocol

Before each experiment session starts the screen position is adjusted to match the participant height: the screen and the cameras are placed at 60 cm away from the participants’ head and just below her line of sight. Each participant goes through the gaze tracker calibration procedure in the presence of the experimenter. When the tracker is correctly calibrated, the experimenter leaves the room and the participant can start the experiment by clicking on a large start button on the screen. The participant is then presented with on-screen instructions about the game controls and proceeds to the tutorial level of the game. After the tutorial, the participant is asked to answer a few demographic questions and, once the questionnaire is completed, she starts to play the three main levels. The total amount of time needed to complete the experiment is, on average, 13 minutes. No time limit is enforced to give the players the possibility to choose the game style they prefer.

²developed by LC Technologies, Inc. - www.eyegaze.com

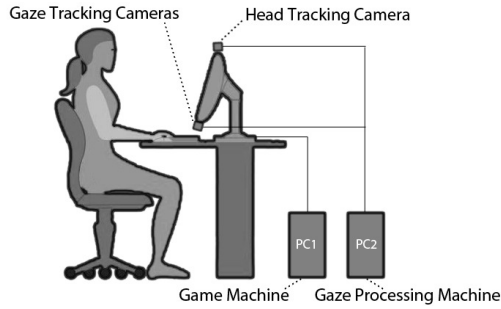


Figure 5: The setup used for our experiments.

4.3 Collected Data

Twenty nine subjects participated in the experiment. Twenty four were male, five were female; the age of the participants ranged between 23 and 34 years. Statistical data about game-play behaviour, virtual camera movement and gaze position is collected for each participant. Data collection is synchronised to the Eyefollower sampling rate, therefore, data is sampled 120 times per second. Each data sample contains:

- Game-play information, including events such as a fuel cell collected, pressed keys and the avatar position.
- Position and orientation of the camera.
- Coordinates of the gaze position on the screen.
- The number and the type of objects around the avatar at each frame classified into two categories: close and far. All the objects reachable by the avatar within the next action are labelled as close otherwise as far.

5. EXTRACTED FEATURES

The data collected for each game is split into three datasets according to the three areas discussed earlier in Section 3.2. Each time a player completes an area two types of statistical features are extracted from that area: game-play and camera related features. The features of the first type are the experiment's independent variables and encapsulate elements of the player's behaviour in the area. The features of the second type describe the camera behaviour for each platform, therefore, they define the experiment's dependent variables.

5.1 Game-play

For each area, the following game-play statistical features have been calculated that reflect the actions performed by each player:

- **Number of fuel cells collected:** the number of collected fuel cells divided by the number of available fuel cells in the area.
- **Number of damage given:** the number of punches given to the enemies.
- **Number of damage received:** the number of times the avatar is hit by an enemy.
- **Number of re-spawn points activated:** the number of re-spawn points activated divided by the number of re-spawn points available in the area.

- **Number of jumps:** the number of jumps performed in the area.

5.2 Camera Behaviour

Camera behaviour can be modelled directly using the data about camera position relative to the avatar. However, this approach would fail revealing which objects the player wants to watch during play. A better approximation would be achieved by analysing the objects present on screen through each area. The presence of a certain object on the screen, however, does not necessarily imply an intentionality of the player; e.g. the object might be on the screen only because it is close to an object the player is interested to. The gaze data available permits to overcome this limitation since, using the gaze position, it is possible to understand which object is actually observed among the ones framed by the player. Therefore, to model the camera behaviour, we combine camera movements and gaze coordinates to identify the objects observed by the player at each frame and we extract the following time-based statistical features:

- **Time watching the avatar:** the time spent watching the avatar divided by the area's completion time.
- **Time watching fuel cells:** the time spent watching fuel cells close to the avatar divided by the time there is at least one fuel cell close to the avatar.
- **Time watching enemies:** the time spent watching enemies close to the avatar divided by the time there is at least one enemy close to the avatar.
- **Time watching re-spawn points:** the time spent watching re-spawn points close to the avatar divided by the time there is at least one re-spawn point close to the avatar.
- **Time watching jump pads:** the time spent watching jump pads close to the avatar divided by the time there is at least one jump pad close to the avatar.
- **Time watching platforms:** the time spent watching platforms close to the avatar divided by the time there is at least one platforms close to the avatar.
- **Time watching far objects:** the time spent watching any object far from the avatar divided by the area completion time.
- **Camera Speed:** the average speed S of the camera relative to the avatar and it is defined as $S = (D_c - D_a)/T$, where D_c is the distance covered by the camera during the completion of an area, D_a is the distance covered by the avatar and T is the time spent to complete the area.

The time spent watching each object is calculated as the sum of the durations of the smooth pursuit and fixation movements of the eyes during which the gaze position falls within the object's projected image. For an object to be watched by the player, it is necessary that this is visible on the screen and thus framed by the camera. Therefore these features are not only directly influenced by the gaze movement but also indirectly by the virtual camera behaviour.

Index	Area Types		
	Collection	Fight	Jump
Davis-Bouldin	3	3	8
Krzanowski-Lai	2	3	3
Calinski-Harabasz	2	3	3
Dunn	2	2	2
Hubert-Levin	3	3	3

Table 2: Number of clusters suggested by each cluster validity index. The selected number of clusters appears in bold.

6. RESULTS

As described earlier in Section 3.2 the extracted features are calculated within each of the areas appearing in a level. Areas missing more than 12 gaze samples (the minimum duration of a fixation movement) are removed from the dataset. Such areas are viewed as experimental noise as they only carry partial information about the observed objects. After this first filtering the dataset contains 1168 records corresponding to all areas completed by the experiment participants with complete gaze information. Although the game contains only 36 areas across the three levels, the player can move freely in each level and thus complete an area more than one time. The area type is labelled dynamically for each record according to the type of game-play available each time, e.g. if a player returns into an area previously labelled as *fight* after having killed all the enemies, the new record will be labelled according to the game-play options the area offers in the second visit. The data is further cleaned by removing all the records (area visits) with a duration lower than 2.5 seconds resulting into 759 records. The records are then sorted into three separate groups according to the area type resulting into three smaller datasets. The collection the fight and the jump area dataset contain 239, 378 and 142 records respectively.

We follow a two-step experimental approach for the analysis of the collected data. As a first step we employ a clustering method on the camera related features of each area type to identify different virtual camera behaviours (see Section 6.1). We then perform a statistical analysis on the relevant differences among these clusters in terms of game-play to find the game-play features that influence camera behaviour (see Section 6.2). We discuss the main findings of this approach in Section 7.

6.1 Clustering

The number of distinct camera behaviours as well as their internal characteristics can only be based, in part, on domain knowledge. One can infer camera behaviour profiles inspired by a theoretical framework of virtual cinematography [2, 13, 10]. or alternatively follow an empirical approach — as the one suggested here — to derive camera behaviour profiles directly from data. The few existing frameworks focus primarily on story-driven experiences with little or no interaction, thus are not applicable in our context. Therefore, we adopt a data-driven approach and we employ the k-means clustering algorithm on the gaze-based extracted features for the purpose of retrieving the number and type of different camera behaviours.

Unsupervised learning allows us to isolate the most significant groups of samples from each dataset. However, k-means requires the number of clusters k existent in the data in or-

Collection Areas ($k = 2$)						
N	AV	FC	JP	RP	DO	SP
150	0.595	0.108	0.034	0.113	0.021	3.338
89	0.361	0.125	0.056	0.072	0.012	8.852

Fight Areas ($k = 3$)							
N	AV	FC	COP	JP	RP	DO	SP
137	0.674	0.042	0.095	0.049	0.034	0.036	3.283
99	0.676	0.032	0.478	0.044	0.056	0.025	5.293
142	0.250	0.029	0.069	0.030	0.052	0.013	5.927

Jump Areas ($k = 3$)					
N	AV	FC	PL	DO	SP
33	0.759	0.464	0.795	0.202	2.1293
80	0.736	0.113	0.658	0.059	2.7593
29	0.450	0.163	0.559	0.012	5.5854

Table 3: Average gaze features values and number of records of each cluster for the three datasets. N is the number of records, AV is the time spent observing the avatar, FC is the time spent observing fuel cells, COP is the time spent observing coppers, JP is the time spent observing jump pads, RP is the time spent observing re-spawn points, PL is the time spent observing platforms, DO is the time spent observing distant objects and SP is the relative camera speed.

der to minimise the intra-cluster variance. To overcome this limitation, the algorithm runs with a progressively higher k value — from 2 to 10 — and the clusters generated at each run are evaluated using a set of five cluster validity indexes. The algorithm runs 50 times for each k and the run with the smallest within cluster sum of squared errors is picked for validation. Table 2 shows the value of k which optimises each validity measure for each dataset, the most appropriate number of clusters for each area type is selected through a majority voting mechanism.

Table 3 shows the average gaze features values of each cluster for all three datasets. The value contained represents the average time spent observing each object divided by the time spent completing the area. It is important to state that the sum of the values of each record can be greater than 1.0 since the player can observe two overlapped objects at the same time. This is possible in a three dimensional game as all the rendered objects have the same distance from the eyes independently from their position in the virtual environment. On the other hand, even if the overall value is greater than 1.0, each record does not represent the complete behaviour of the gaze but it focuses only on the task-relevant objects.

The two clusters emerging from the gaze behaviour data of the collection areas show two very distinct behaviours in terms of camera speed and time spent observing the avatar and the fuel cells. The first cluster of players spend, on average, almost twice as much time as the players of the second cluster observing the avatar. Moreover, the players belonging to the first cluster spend on average 20% less time observing fuel cells and move the camera at less than half of the speed, compared to the players belonging to the second cluster. These values suggest that the camera for the first cluster should move slowly and focus primarily on the avatar, while the behaviour exhibited by the second cluster

Area Type	Fuel Cells	Damage Given	Damage Received	Re-spawn Points	Jumps
Collect.	5.02	-	-	1.23	0.76
Fight	1.63	12.42	10.89	0.49	9.64
Jump	11.98	-	-	-	0.53

Table 4: F-distribution values of the inter-cluster ANOVA test on the game-play features. The threshold for a 5% significance level is $F > 3.85$ for the collection and fight areas and $F > 2.99$ for the jump areas. Values above the threshold appear in bold.

would be better modelled with a fast camera with an equal focus on the avatar and the other relevant items in the area. We label these behaviour patterns respectively as *Slow On Avatar* and *Fast On Task*.

In the jump areas, as expected, all the players spend a large amount of time observing the platforms; however, their camera behaviour differs on all the other parameters. The first cluster evenly focuses on all the objects in the area and moves the camera at very low speed (2.1293). Such values suggest that the camera for this cluster should stay at a long distance in order to frame all the objects of each platform and should move at a low speed. We label this behaviour as *Slow Overview*. Both the second and the third clusters of players focus primarily on the avatar and the platforms, but differ on the camera speed. Since the platforms are the primary task objects in these areas, we label the two clusters respectively as *Slow On Task* and *Fast On Task*.

The first and the second cluster of the fight areas exhibit a behaviour matching, respectively, the *Slow On Avatar* and the *Fast On Task* patterns. The third cluster of players, on the other hand, observe all the object for a very short time suggesting far camera moving at high speed. We label this behaviour as *Fast Overview*.

6.2 Statistical Analysis

The areas have been categorised into three types to sort out the trivial differences related to the challenges available in each area. However, within each area type the players have the full authority to chose which action to perform, e.g. in a fight area a player could fight each enemy or just run through the area without engaging in any combat. For this reason, it is necessary to identify the differences between the clusters identified in terms of player behaviour.

The null hypothesis in the statistical test employed is that there is no statistical difference between the game-play features of the various clusters identified by k-means. An inter-cluster one-way ANOVA is performed for each game-play feature to identify for which features we can reject the null hypothesis and, therefore, isolate the gameplay features that have an effect to the camera behaviour.

Table 4 shows the results of the test. Values greater than 3.85 for the collection and fight areas and 2.99 for the jump areas identify the features which exhibit a significant inter-cluster difference with a 5% significance level. In the fight areas dataset there is a significant difference in terms of damage (both given and taken) and number of jumps. In the other two area datasets the clusters differ significantly in the number of fuel cells collected.

The camera behaviour differences between the two clusters of the collection areas emerge also in game-play. The

Collection Areas		
Fuel Cells	Re-spawn Points	Jumps
0.526	0.001	2.719
0.624	0.000	2.774

Fight Areas				
Fuel Cells	Damage Given	Damage Received	Re-spawn Points	Jumps
0.148	0.788	0.759	0.088	2.416
0.179	2.909	1.747	0.136	0.798
0.163	0.923	0.535	0.127	1.606

Jump Areas	
Fuel Cells	Jumps
0.893	4.065
0.558	4.300
0.606	4.034

Table 5: Average game-play features values of each cluster for the three datasets. Significantly different values appear in bold.

average number of collected items of the two clusters is significantly different and, as shown in Table 5, the players of the first cluster collect on average about 20% less fuel cells than the second cluster. Similarly, the three clusters of the jump areas are characterised by a different average number of collected items. The first cluster of players collect, on average, approximately 50% more fuel cells compared to the second and 60% more compared to the third.

In the fight areas the playing clusters differ in the amount of damage received, damage given and number of jumps. The first cluster of players is identified by little and ineffective interaction with coppers (both damage taken and given have equal values) and a high number of jumps. The third cluster of players shows a similar attitude towards coppers as the damage given value is rather similar to ones in the second cluster. However, the third cluster of players show a much lower amount of damage taken (about 40% lower) and almost half the number of jumps, demonstrating a better attitude towards fighting. The second cluster is characterised by high levels of damage taken and given and a number of jumps considerably smaller than the other two clusters.

7. CONCLUSIONS

This paper proposed an approach to model players' camera control behaviour and its relationship to game-play and the in-game player behaviour. The camera behaviour models are based on information about players' gaze and virtual camera position collected during a game experiment. The data collected is split into small areas with coherent game-play and then categorised into three groups according to the game-play provided to the player. For each group the data is clustered using k-means to identify relevant behaviours and the clusters are analysed for significant differences in terms for game-play experience. The clustering of the data reveals 2 clusters of camera behaviour for the *collection* areas and 3 clusters for camera behaviour for the *fight* and *jump* areas, each characterized by a specific pattern of control. Moreover, the analysis of variance of the extracted game-play features between the clusters shows a significant relation-

ship between the clusters and some game-play features.

Such relationship could potentially be exploited to provide a personalised camera behaviour. For instance, in the fight areas, an automatic camera control system might be instructed to generate a *Slow On Avatar* camera if the player tends to perform a high number of jumps, while it might be instructed to generate a *Fast Overview* camera if the player tends to fight often with the enemies present on the platforms. Moreover, by following the approach proposed in this article, the same principle could be applied in any game incorporating a third person camera, enabling the game designers to create more personalised visual experiences.

The analysis of collected data also suggests that the game is visually very complex and a multitude of objects compete for the player's attention. This is especially evident in the camera behaviour clusters in which the players do not focus primarily on the task-related objects but spend a large amount of time observing the virtual environment. This aspect could be limited by reducing the visual noise in the game, but this would require to turn down the game and drastically reduce the applicability to other games of the same genre. The actual version of the game incorporates visually and in terms of game-play all the standard features of modern 3D platformers and, besides the aforementioned limitations, the results give an evidence of the relationship between camera behaviour models and game-play in such games. Moreover the same methodology could be used to build camera behaviour models for different games such as role-playing games or action games.

As a future step, a precise model of the relationship between game-play and camera behaviours could be built from the collected data in order to dynamically recognise the game play pattern and apply the right camera behaviour model. Moreover, the model could be adapted in real-time to optimise one or more aspects of the player experience with the introduction of an affective model of camera behaviour.

8. ACKNOWLEDGMENTS

The authors would like to thank Fiona Mulvey for her precious support on gaze-tracking and all the anonymous participants of the experiment.

9. REFERENCES

- [1] D. Amerson, S. Kime, and R. M. Young. Real-time cinematic camera control for interactive narratives. In *ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, Valencia, Spain, 2005.
- [2] W. H. Bares and J. C. Lester. Cinematographic User Models for Automated Realtime Camera Control in Dynamic 3D Environments. In *International Conference on User Modeling*, 1997.
- [3] W. H. Bares, S. McDermott, C. Boudreaux, and S. Thainimit. Virtual 3D camera composition from frame constraints. In *ACM international conference on Multimedia*, 2000.
- [4] W. H. Bares, L. S. Zettlemoyer, D. W. Rodriguez, and J. C. Lester. Task-Sensitive Cinematography Interfaces for Interactive 3D Learning Environments. In *International conference on Intelligent user interfaces*, 1998.
- [5] M. Bernhard, E. Stavrakis, and M. Wimmer. An empirical pipeline to derive gaze prediction heuristics for 3D action games. *ACM Transactions on Applied Perception (TAP)*, 2010.
- [6] J. Blinn. Where Am I? What Am I Looking At? *Computer Graphics and Applications, IEEE*, 1988.
- [7] O. Bourne and A. Sattar. Applying Constraint Weighting to Autonomous Camera Control. In *Artificial Intelligence In Interactive Digitale Entertainment Conference*, 2005.
- [8] F. Charles, J.-l. Lugin, M. Cavazza, and S. J. Mead. Real-time camera control for interactive storytelling. In *Game-On*, 2002.
- [9] D. Christianson, S. Anderson, L.-w. He, D. Salesin, D. Weld, and M. Cohen. Declarative Camera Control for Automatic Cinematography. In *AAAI. AAI*, 1996.
- [10] M. Christie and J.-M. Normand. A Semantic Space Partitioning Approach to Virtual Camera Composition. *Computer Graphics Forum*, 24, 2005.
- [11] M. Christie, P. Olivier, and J.-M. Normand. Camera Control in Computer Graphics. In *Computer Graphics Forum*, 2008.
- [12] S. M. Drucker and D. Zeltzer. Intelligent camera control in a virtual environment. In *Graphics Interface*, 1994.
- [13] A. Jhala and R. M. Young. A discourse planning approach to cinematic camera control for narratives in virtual environments. In *AAAI*, 2005.
- [14] H. P. Martinez, A. Jhala, and G. N. Yannakakis. Analyzing the impact of camera viewpoint on player psychophysiology. In *International Conference on Affective Computing and Intelligent Interaction*, 2009.
- [15] D. Pinelle and N. Wong. Heuristic evaluation for games. In *CHI Conference on Human Factors in Computing Systems*, 2008.
- [16] V. Sundstedt, E. Stavrakis, M. Wimmer, and E. Reinhard. A psychophysical study of fixation behavior in a computer game. In *Proceedings of the 5th symposium on Applied perception in graphics and visualization*, 2008.
- [17] C. Ware and S. Osborne. Exploration and virtual camera control in virtual three dimensional environments. *SIGGRAPH*, 1990.
- [18] G. Yannakakis, H. Martinez, and A. Jhala. Towards Affective Camera Control in Games. *User Modeling and User-Adapted Interaction*, 2010.
- [19] A. L. Yarbus. *Eye movements and vision*. Plenum press, 1967.