User Modelling and User-Adapted Interaction manuscript No.
(will be inserted by the editor)

Adapting Virtual Camera Behaviour through Player Modelling

Paolo Burelli and Georgios N. Yannakakis

Received: 19 December 2013 / Accepted: 14 January 2015

Abstract Research in virtual camera control has focused primarily on finding methods to allow designers to place cameras effectively and efficiently in dynamic and unpredictable environments, and to generate complex and dynamic plans for cinematography in virtual environments. In this article, we propose a novel approach to virtual camera control, which builds upon camera control and player modelling to provide the user with an adaptive point-of-view. To achieve this goal, we propose a methodology to model the player's preferences on virtual camera movements and we employ the resulting models to tailor the viewpoint movements to the player type and her game-play style. Ultimately, the methodology is applied to a 3D platform game and is evaluated through a controlled experiment; the results suggest that the resulting adaptive cinematographic experience is favoured by some player types and it can generate a positive impact on the game performance.

Keywords Virtual Camera Control · Gaze Interaction · Player Modelling · Computer Games

1 Introduction

The virtual camera provides the principal means of interaction between a user and a virtual environment as it represents her point-of-view. Aspects such as camera placement (i.e. the configuration of the camera parameters within the virtual environment) and camera animation (i.e. the process of transitioning from one set of camera parameters to another one) play a vital role in 3D interactive environments and have an important impact on usability and the overall user experience (Pinelle

Paolo Burelli
Department Of Architecture, Design and Media Technology
Aalborg University Copenhagen, Denmark
E-mail: pabu@create.aau.dk

Georgios N. Yannakakis
Institute of Digital Games, University Of Malta, Msida, Malta
E-mail: georgios.yannakakis@um.edu.mt

et al., 2008). In particular, in 3D computer games, the presentation of the game events largely depends on the camera position and movements; thus, virtual cinematography has a deep impact on both gameplay and storytelling as a whole.

In computer games, the virtual camera is usually either controlled by the player or predefined by a game designer. These two approaches, often not mutually exclusive, have different advantages and disadvantages.

One one hand, if the camera is under player control, while the player's control is often assisted by the game, the complexity of the interaction with the camera can be very high, reducing the ability of the player to interact with the environment; furthermore, it reduces the designer's control over game storytelling.

On the other hand, designer-placed cameras release the player from the burden of controlling the point of view but cannot guarantee a correct visualisation of all possible player actions, often leading game designers to reduce the range of possible player actions to be able to generate a more cinematographic player experience. Moreover, in multi-player games or in games where the content is procedurally generated (Togelius et al., 2011), the designer has potentially no information to define *a priori* the camera positions and movements.

The problem of defining the best camera control paradigm has been investigated by a number of researchers over the last three decades (Christie et al., 2008) and the current predominant line of research aims to find effective and efficient algorithms to automatically place and animate the camera given some high-level and environment-independent requirements. Within this paradigm, an optimal camera configuration is defined as the combination of camera settings that maximise the satisfaction of the requirements imposed on the camera, often called a *camera profile* (Bourne et al., 2008).

These requirements describe the desired camera behaviour in terms of abstract properties of frame composition and motion. Frame composition describes the way in which the objects should be framed by the camera, such as their position in the frame or the size of their projected image. Camera motion requirements describe the way the virtual camera should be animated in the virtual environment while framing the subjects. The definition of these requirements originates from the rules used to shoot real-world scenes in cinematography and photography (Arijon, 1991).

Even though the definition of the requirements is commonly delegated to human designers who hand-craft the cinematographic experience, some efforts have been dedicated also to the automation of this process (Bares and Lester, 1997a; Bares et al., 1998; Tomlinson et al., 2000). In particular, the studies of Bares et al. (1997a; 1998) have investigated the personalisation of the cinematographic experience through task and user modelling. We follow a similar direction with the objective of closing the control loop of automatic camera control; contrarily however to previous studies we aim to allow the player to affect the automatic camera controller implicitly. Our hypothesis is that allowing the player to implicitly control the camera would simplify the interaction and lead the player to be more satisfied and feel more engaged in the game.

For this purpose, we have investigated player preferences concerning virtual camera placement and animation through a study in which we analysed the behaviour of a group of players while playing a platform game. During their playing experience, we recorded the way they controlled the virtual camera and their locus of attention on the screen through gaze tracking, and aggregated this information

to identify their camera behaviour patterns (Picardi et al., 2011). Based on the results of this study, we developed a neural-network based model of the relationship between camera behaviour patterns, player behaviour and game-play context (Burelli and Yannakakis, 2011). This model is designed to be employed to drive an automatic camera controller (as introduced by Burelli and Yannakakis (2010)) and provide a personalized camera experience.

In this article, we gather the results of the our previous studies, we abstract a coherent methodology and we complete it by describing how to connect camera behaviours and automatic camera control. Furthermore, we showcase a complete application of the methodology in a commercial-grade three-dimensional computer game. Finally, we conclude this study on adaptive camera control by presenting a user evaluation and a discussion of its results, which show that the personalised camera experience had a positive impact on the players' performance and was perceived extremely positively by non-expert participants.

After a presentation of the related works in the area in Section 2, we will present the general methodology in Section 3 and the game used to test it in Section 4. In Section 5 and 6, we will report the results of the previous works on the modelling phase of the methodology and we will present the application of the adaptation phase. A user evaluation of the resulting adaptive experience will be presented in Section 7 and the results will be discussed in Section 8.

2 Background

Since the introduction of three-dimensional computer graphics, virtual camera control attracted the attention of a large number of researchers (refer to (Christie et al., 2008) for a comprehensive review). Early approaches focused on the mapping between the degrees of freedom (DOF) for input devices and 3D camera movement (Ware and Osborne, 1990) for direct camera manipulation; however, direct manipulation soon demonstrated to be problematic for the user, driving researchers to investigate how to simplify camera control by automatising the camera placement and animation (Phillips et al., 1992; Drucker and Zeltzer, 1994). The first example of an automatic camera control system was showcased by Blinn (1988). He designed a system to automatically generate views of planets in a space simulator for NASA. Although limited in its expressiveness and flexibility and suitable only for non interactive applications, Blinn's work served as an inspiration to several other researchers that investigated more efficient solutions and more flexible mathematical models able to handle more complex aspects such as camera motion and frame composition (Arijon, 1991) in static and dynamic contexts.

Automatic camera control identifies the process of automatically configuring the camera in a virtual environment according to a set of requirements. It is a non-linear automatic control problem (Pontriagin, 1962): the system's input are the requirements on composition and motion, while the internal state of the system is defined by the camera parameters — e.g. position and rotation. Following this model, it is possible to identify three main research questions:

- How to find the optimal state of the camera with respect to some given requirements.
- How to control the dynamic behaviour of the camera.
- How to identify the best inputs for the system.

The first question attempts to explore the appropriate approach to be used to find the optimal camera configuration that satisfies the input requirements. Constraint satisfaction or optimisation techniques are often used for this purpose. The second question revolves around the control of the camera movements during the framing of one scene and during the transitions between two subsequent shots. The last questions is often called camera/shots planning and is associated with the selection of the shot type to be used to frame a certain event in the virtual world — i.e. the automatic control system's inputs. It is within this area of camera control, that we propose a methodology to adapt the inputs of an automatic camera controller to the way the player plays.

Camera planning was defined for the first time by Christianson et al. (1996) as the problem of automatically scheduling the sequence of shots to film one or more events in a virtual environment. In their work, Christianson et al. proposed a declarative language to describe the behaviour of the camera in terms of a linear sequence of idioms. He et al. (1996) extended their work by allowing the script to define more complex behaviours incorporating conditions through a finite state machine, which was extended further by Jhala and Young (2005) by employing a planning approach to automatically select a shot given a story. Opposed to the aforementioned top-down approaches, Tomlinson et al. (2000) proposed a bottom-up approach to shot definition, in which the camera is modelled as an autonomous virtual agent, called *CameraCreature*, employing an affective model and a set of motivations. The agent shoots the most appropriate shot at every frame according to the events happening in the environment and its current internal state.

The work of Tomlinson et al. (2000) departs from the idea of a designer defined cinematographical experience; however, the system is unaware of its impact on the user experience and the user is unable to influence the behaviour of the camera. Bares and Lester researched this aspect by suggesting and evaluating a system that selects the most appropriate camera settings depending on the user's tasks in a virtual learning environment (Bares and Lester, 1997b; Bares et al., 1998). Furthermore, they investigated the idea of modelling the camera behaviour according to the user preferences to generate a personalised cinematographic experience (Bares and Lester, 1997a). The user model construction required the user to specifically express some preferences about the style for the virtual camera movements.

Yannakakis et al. (2010) studied the impact of camera viewpoints on player experience and built a model to predict this impact, demonstrating the existence of a relationship between player emotions, physiological signals and camera parameters. However, since the relationship was built on low-level camera parameters, the findings give limited information about which visual features are more relevant for the player. Burelli (2013) has further investigated this relationship by analysing the impact of different shot compositions on player experience. The results show that, depending on the task performed by the players, visual aspects, such as symmetry or spacing, are highly correlated to players' arousal. These results suggest that employing directly low-level camera behaviour features — e.g. displacement or height — to construct reliable computational models of a virtual camera's behaviour is not possible as it is necessary to understand what is the visual content which is reproduced on screen by the camera and which, among the elements visualised, are intentionally placed on screen and are therefore relevant to the player.

One possible method to extrapolate information about the above aspects has been employed by Sundstedt et al. (2008), who conducted an experimental study to analyse players' gaze behaviour during a maze puzzle solving game. The results of their experiment show that gaze movements, such as fixations, are heavily 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. (2010) performed a similar experiment using a three-dimensional first-person shooter game in which the objects observed by the players were analysed to infer the player's level of attention. Similarly, El-Nasr and Yan (2006) used an eye tracker to record eye movements during a game session to determine eye movement patterns and areas of interest in the game.

The results of these studies reveal the potentials of gaze tracking as means of identifying visually relevant features; following this idea, we investigated the possibility to employ players' gaze to describe the camera behaviour based on the visualised content instead of the camera's low-level parameters (Picardi et al., 2011; Burelli and Yannakakis, 2011). In this paper we propose and evaluate a general methodology to generate personalized cinematographic experiences in computer games. The methodology is used to build personalized camera behaviour profiles for players playing a 3D platform game; the profiles are then used to drive an automatic camera controller (Burelli and Yannakakis, 2010) and the resulting cinematographic experience is evaluated through a user experiment.

3 Methodology

Tomlinson et al. (2000), in their paper on expressive autonomous cinematography, quote a statement by Steven Drucker at SIGGRAPH '99: "It was great! I didn't notice it!''. Drucker was commenting Tomlinson's work presented at SIGGRAPH that year and, in his comment, he clearly associates the quality of a camera control system with the lack of intrusiveness; however, how to achieve such a result and how to take the control of the camera from the player while still matching the player expectations are open research questions. We believe that, to bridge the gap between automatic and manual camera control, the camera behaviour should be affected by the player; in our view, the camera control system should be able to learn camera preferences from the user and adapt the camera profile to improve the player experience.

To achieve this goal while still holding the advantages of an automatically controlled camera, it is necessary to identify a way for the camera to adapt its behaviour to match the player's preferences. As a first step towards this goal, we need to investigate a method to understand player preferences concerning virtual camera placement and animation. Furthermore, to apply such preferences during a game-playing session, we need to model the relationship between camera behaviour, player behaviour and game context, so that we can use the latter two to predict the first one and then, instruct the virtual camera accordingly.

In this context, player behaviour describes the way the player performs in the game — e.g. how many jumps she performs, while camera behaviour describes how a player moves the camera and what she would like to frame with it. In the proposed protocol, we suggest that a number of players should play a version of the

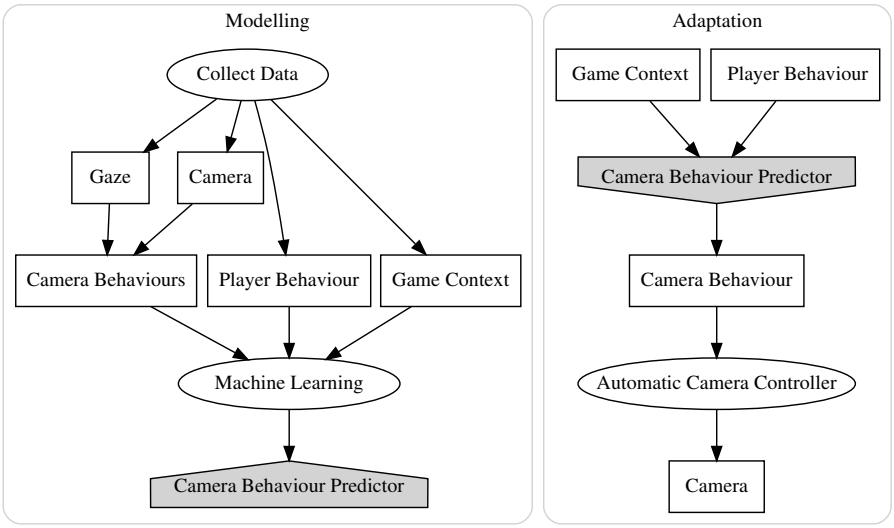


Fig. 1: Camera behaviour modelling and adaptation phases of the adaptive camera control methodology. The first phase (left) leads to the construction of a camera behaviour model, which is used in the adaptation phase (right) to drive the automatic camera controller and generate the adaptive camera behaviour.

game in which they directly control the camera; during this session, parameters about the players' gaze, camera movement, players' actions and game context are recorded. From this data, the camera behaviour is modelled on a combination of gaze and direct camera position information. The resulting data is analysed to identify different virtual camera behaviour patterns. Finally, the relationship among these patterns, player behaviour and game context behaviour is modelled using machine learning. In the second phase of the methodology we propose, these models are used to identify the best camera behaviour pattern at each point of the game and use it to drive an automatic camera controller; thus, generating personalised virtual cinematographic experiences.

The two aforementioned phases of the methodology are depicted in Figure 1: the first phase is required to learn a camera behaviour model for a game (left diagram) which is later used to select the most appropriate camera behaviour for the player when the game is played (right diagram). The remainder of this section describes, in general terms, the different steps of the methodology proposed, while the following sections describe an application of the methodology with details on the algorithms and the features used for each step.

3.1 Modelling

The first phase of the proposed methodology requires modelling the behaviour of the camera and finding a relationship with the player behaviour in the games.

This relationship can be exploited while the game is played to activate the correct camera behaviour to support the player's action. The simplest way to model the camera's behaviour consists of directly using data about camera position relative to the avatar. This approach was employed by Yannakakis et al. (2010) to investigate the relationship between player experience and camera placement; however, a later study by Burelli (2013) showed that such a relationship can be more accurately modelled by analysing the content visualised by the camera rather than its position or angle.

In the proposed methodology, we follow the latter direction and use the objects visualised by the camera as a way to describe its behaviour. However, when manually controlling a camera, the presence of a certain object on the screen 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. In one of the authors' previous study (Picardi et al., 2011), such a problem was tackled using player's gaze position to understand which object is actually observed among the ones framed by the player. Based on the results of that study, we propose to model the camera behaviour on a combination of camera movements and gaze coordinates to identify the objects observed by the player at each frame. To generate the camera behaviour model for a specific game, we propose to collect the aforementioned data regarding the players' camera behaviour while playing the game using a manual camera controller (e.g. during an early phase of the game testing). Figure 2 shows an example of the setup needed during the data-collection phase: the experiment participant plays the game at a desk sitting on a chair with rigid back and no wheels (to reduce head movements), the gaze is tracked using an infra-red camera to detect and track eye features. Subsequently, in line with the approach described by Mahlmann et al. (2010), we mine the logged data to identify a set of prototypical behaviours and then use machine learning to build a model connecting the player in-game behaviour with the camera behaviour types.

3.1.1 Camera and Player Behaviour

To describe the camera behaviour we analyse the content visualised by the players on the screen. For this reason, we use gaze position on the screen to perform a ray-casting from the camera position toward the gaze direction and check which object is hit by the ray. This information is combined with information about the player's eye movements to identify *fixations* and *smooth pursuits* (Yarbus, 1967). As a result, we can identify which objects in the virtual environment have been observed and for how long. Fixations and smooth pursuit movements are not necessarily accurate indicators of cognitive processing, since a person is able to process visual information also surrounding the gaze position and the gaze itself can be driven subconsciously towards salient image areas without an active cognitive process. However, in a visually rich virtual environment such as a computer game, the locus of attention coincides with the gaze direction (Irwin, 2004); moreover, due to the task-oriented nature of games, fixations and pursuits are mainly attention-driven (Land, 2009).

From the fixation and smooth pursuit movements recorded, a vector of features is extracted in which each feature represents the amount of time the player spends framing and observing each object type in the game environment. The length of the features vector depends on the number of different objects types present

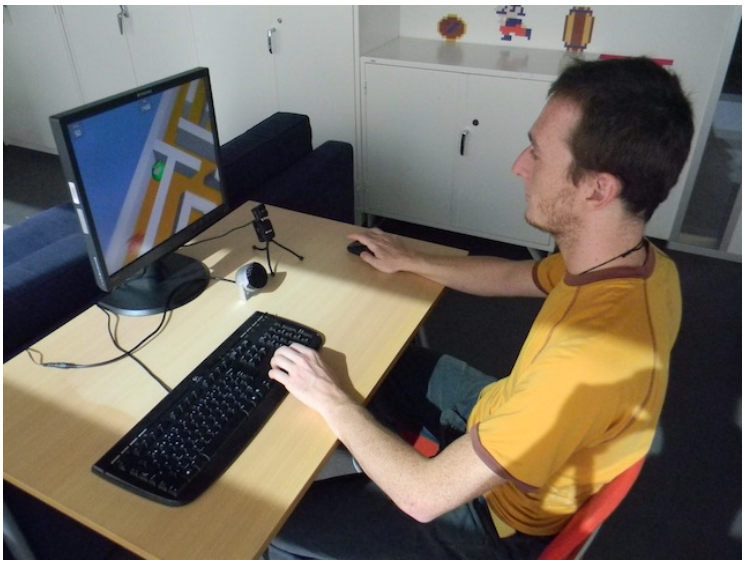


Fig. 2: Example of a data collection setup. The player plays a game and manually controls the virtual camera. The game keeps track of the virtual camera placement, the player behaviour and the gaze position on the screen

in the game. The time 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 an object's projected image. The vector can be calculated, depending on the mechanics of the game, either over the course of the whole game session or once for each game segment; the latter option is preferable as it leads to multiple, game context dependent, camera behaviours.

The granularity of the subdivisions in segments depends on the game characteristics. Three different criteria could guide the subdivision process: time, space and task. According to the task criterion, the logged data for each player is divided in records representing the experience for each task completed by the player. While this solution is probably the most appropriate, as it is based on game characteristics, it is often inapplicable as some parts of the game might have unclear tasks or multiple ones active at the same time. The space criterion is simpler to apply as in many games there are well defined areas, e.g. *Devil May Cry* (Capcom, 2001) or *Halo* (Microsoft, 2001); therefore, it is easy to identify the beginning and the end of each record. In case neither of the previous criteria can be applied, the overall experience can be divided in time segments of variable length depending on the game characteristics. The observation times describing the gaze behaviour should be normalised for each record segment on the segment's length.

From the vectors describing the camera behaviour in each segment, we need to identify which and how many camera behaviour types exist and what are their internal characteristics. Such analysis can be partially based on domain knowledge: one can infer camera behaviour profiles inspired by a theoretical framework of virtual cinematography (Jhala and Young, 2005). However, the few existing frameworks focus primarily on story-driven experiences with little or no interac-

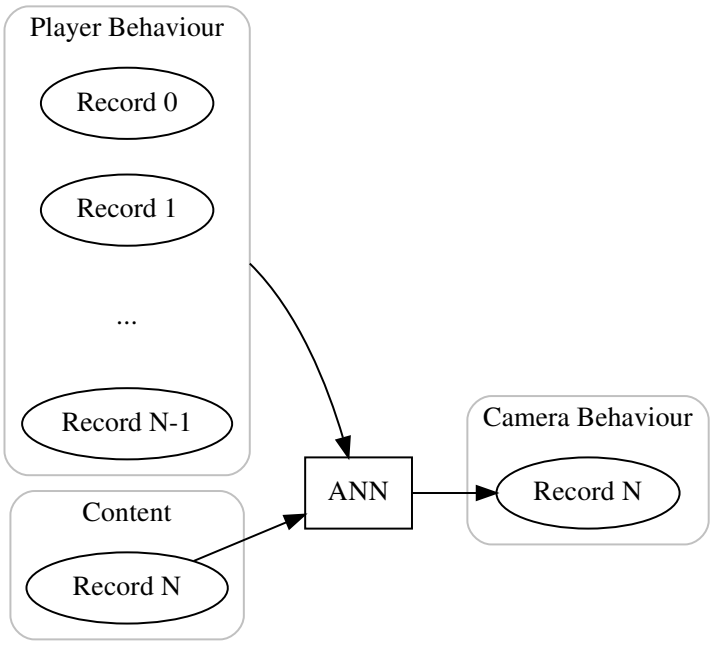


Fig. 3: Camera behaviour prediction scheme. The artificial neural network is presented with the features describing the gameplay characteristics of the next record and the features about the previous player behaviour as input and returns the next predicted camera behaviour for the next record as the output.

tion, thus are not applicable in our context. Therefore, we propose to employ a data-driven approach — i.e. derive camera behaviour profiles directly from the collected feature vectors — employing clustering to retrieve the number and types of different camera behaviours. This results to a number of prototypical camera behaviours, with different characteristics in terms of objects to be framed and movement speed.

Similarly to what is performed for the camera, the player behaviour should also be described in quantitative terms as a vector of features for each segment considered. This information is necessary to find a model of the relationship between camera behaviour and player behaviour that can be used, during the adaptation phase, to give the player the appropriate camera behaviour. This vector could contain statistics such as the average health of the avatar or the number of falls; however, the number and type of the features describing the player behaviour varies depending on the game. And, depending on the complexity and variety of the game mechanics, the records might be sorted into different groups according to the type of interaction. Section 5 will present an example of the application of this methodology to a platform game with a description of the concrete feature vectors describing both camera behaviour and player behaviour.

3.1.2 Camera Behaviour Prediction

Once the camera behaviour patterns are identified, we model the relationship between player behaviour and camera behaviour types. More precisely, since the model is intended to select the most appropriate camera behaviour that fits the player's preferences in the game, we attempt to approximate the function that maps the game-play behaviour of the player to the camera behaviour. For this purpose, we have used Artificial Neural Networks (ANNs) (Burelli and Yannakakis, 2011), which have been chosen as a function known for its universal approximation capacity. ANNs and back propagation (Rumelhart et al., 1986) can be used to learn the connection between the player in-game behaviour and its camera behaviour; however, to be able to adapt the camera behaviour, the network needs to be trained in predicting the future player's choice of camera profile, so that the automatic camera controller can be instructed to generate the desired camera behaviour exactly at the moment the behaviour is needed. For this purpose, the ANN is trained to learn the association between the camera behaviour cluster at the record i and the player's in-game behaviour in the previously recorded records (from 0 to $i-1$). Moreover, if the game includes different types of challenge, the ANN should also consider the characteristics of the game-context at segment i (see Figure 3).

3.2 Adaptation

Using this model of camera behaviour, it is possible to predict the player's preferences of camera profile during the game and use this information to instruct an automatic camera controller and personalise the cinematographic experience. When the player performs a switch from one segment (area, task or time frame, depending on the subdivision criterion) to the next one, the prediction of the neural network can be used to decide the right camera behaviour for the next segment of the game.

Once this is selected, a camera profile — i.e., a set of frame and motion constraints — should be generated; this process can be approached in two ways: either a designer assigns a custom designed camera profile to each behaviour identified by the clustering algorithm — which is then automatically picked by the adaptation mechanism — or the profile is generated completely based on the selected behaviour's characteristics. Independently of the approach chosen, the translation process between gaze-based camera behaviours and camera profiles is hardly generalisable over different games as the number and the quality of the objects present on screen vary considerably.

In most action games — e.g. *Tomb Raider* (Eidos Interactive, 1996) or *Super Mario 64* (Nintendo, 1996) — the camera can be instructed to follow the main avatar and maintain the visibility of the objects that have received visual attention in the camera behaviour. The weights of the frame constraints imposed on each object can be related to the amount of time spent observing the objects of the same kind as this information is related to the amount of attention that object receives.

Such an approach can be applied to virtually any game that features an avatar and a third person camera and the constraints imposed on the avatar and on the other objects included in the behaviour can be changed to alter the overall

(a) (b) (c) (d) (e) (f)

Fig. 4: Main components of the Lerpz Escape game. From left to right: player's avatar (Lerpz), a platform, a collectible item (fuel canister), an enemy (copper), a respawn point and Lerpz's spaceship.

composition. For different game genres, such as strategy games or racing games, different strategies can be developed to transform behaviours into camera profiles.

Once a camera profile is generated, it is employed to instruct an automatic camera controller, a software capable of dynamically animating the camera based on a set of frame and motion requirements (Christie et al., 2008). Different approaches have been proposed to solve the camera placement and animation task ranging from gradient descent (Bourne et al., 2008; Lino and Christie, 2012) to population based optimisation (Burelli and Yannakakis, 2010; Ranon and Urli, 2014). The methodology introduced in this article can be applied to any of these, with minimal adjustments depending on the number and type of frame constraints supported.

4 The Game

To showcase the applicability of the proposed methodology and the effects of adaptive camera movements on the player, we applied the methodology to a 3D platform game on which we built a set of models of camera behaviour and implemented the proposed adaptation mechanism.

The game is a custom version of Lerpz Escape, a tutorial game by Unity Technologies[1] and it features an alien-like avatar (Lerpz, see Figure 4a) trapped in a futuristic 3D environment made of floating platforms (see Figure 4b). The platforms can be flat, or be composed of multiple smaller platforms with different heights clustered together. Each platform can be connected to another platform through a bridge or can be disconnected; in which case, the avatar is required to jump to move from one platform to the other.

The game includes three main elements/objects, which the avatar can interact with: *fuel canisters*, *coppers* and *re-spawn points*. Fuel canisters (see Figure 4c) are floating items that the player needs to collect to complete the game. Coppers (see Figure 4d) are animated robots that chase the avatar and hit it until it 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 allows the player to collect one

[1] http://www.unity3d.com

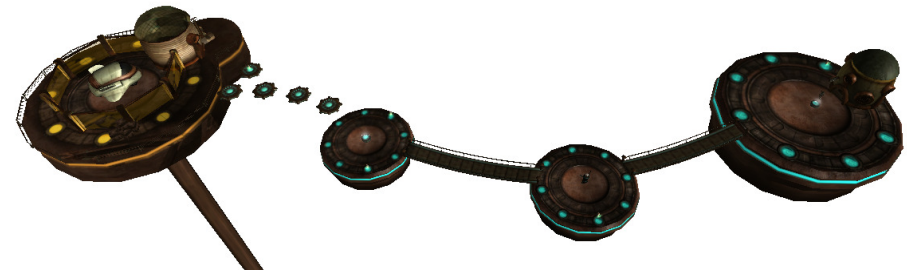
Fig. 5: User interface of the modified version of Lerpz Escape employed in the study. The text in the lower left corner explains the objective of the player. The number in the upper left corner shows the time elapsed since the beginning of the playing session, while the one on the upper right corner shows the number of fuel canisters collected. In the bottom right corner, the control scheme is explained.

extra fuel canister that spawns out of the robot's wreckage. Re-spawn points (see Figure 4e) are small glowing 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.

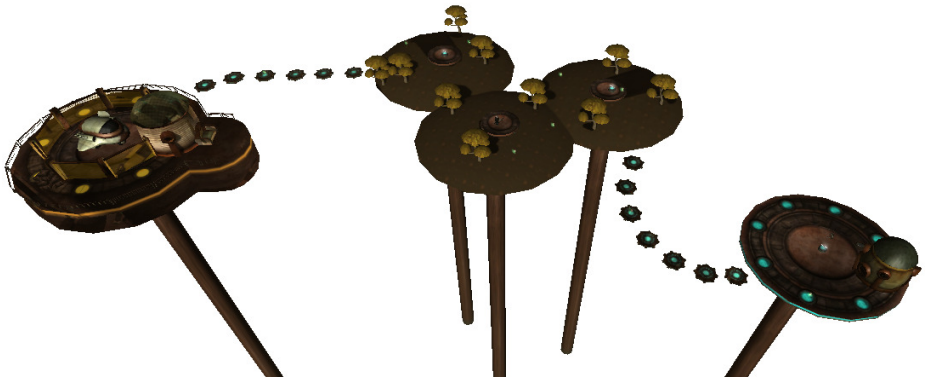
As shown in Figure 6, the game consists of two stages: a tutorial stage and the main stage; each stage starts with the avatar standing on the initial re-spawn point, and ends when the avatar reaches the spaceship on the last platform. The tutorial stage includes a walk-through of the game controls and an explanation of the purpose and the effects of the different objects available. Moreover, during the tutorial the player is presented with all the four main challenges she will face during the main game: jumping from platform to platform, facing an enemy (i.e. a *copper*), collecting items (i.e. *fuel canisters*) and reaching the end of the stage (i.e. the *spaceship*).

The goal of the player is to control Lerpz and make it reach its spaceship (see Figure 4f) at end of the stage. However, the spaceship is surrounded by a force field that does not allow any access. To deactivate the field, Lerpz has to collect a number of fuel canisters (2 for the tutorial stage and 6 for the main stage) across the virtual environment. Moreover, the task has to be performed within a predefined time window, otherwise the current stage will stop and the game will proceed to the next stage whether the goal has been reached or not.

The player has direct control of the avatar and the camera. The avatar is controlled using the keyboard directional keys and the movements are defined in a



(a) Tutorial Stage



(b) Main Stage

Fig. 6: The two stages composing the game used in the case study

camera-relative space, therefore the avatar will move left, right, forward and backward relatively to the player's point of view, which follows the character pointing always at it. The avatar can jump and punch by pressing the Z and X keys. The description of the controls, the objective, the number of collected items and the time remaining are displayed on screen (see Figure 5).

The game designed for this evaluation stands as a prototypical platform game, as it incorporates the typical aspects of the genre. According to the classification described by Wolf (2001), platform games are defined as *"games in which the primary objective requires movement through a series of stages, by way of running, climbing, jumping, and other means of locomotion"*. Moreover, according to Wolf, such games often *"involve the avoidance of dropped or falling objects, conflict with (or navigation around) computer-controlled characters, and often some character, object, or reward at the top of the climb which provides narrative motivation"*. Therefore, we can reasonably assume that the results of the evaluation performed using the aforementioned game are generalizable to the whole *platform games* genre and, at least partially, to genres such as Adventure and Obstacle Course which have a very similar interaction scheme.

In the tutorial stage the players have to face one copper, collect at most five fuel canisters and activate two respawn points. Only one copper is present in the main stage, while fuel canisters and respawn points are 10 and 3, respectively. The

Cl.	Avatar	Fuel Canis.	Coppers	Re-spawn Points	Distant Objects	Platf.	Speed	Label
C1	59.5%	10.8%	-	11.3%	2.1%	-	3.34	slow self-cent.
C2	36.1%	12.5%	-	7.2%	1.2%	-	8.85	fast commit.
J1	75.9%	46.4%	-	-	20.2%	79.5%	2.13	slow detached
J2	73.6%	16.6%	-	-	5.9%	65.8%	2.76	slow commit.
J3	45.0%	11.3%	-	-	1.2%	55.9%	5.59	detached
F1	67.4%	4.2%	9.5%	3.4%	3.6%	-	3.28	slow self-cent.
F2	67.6%	3.2%	47.8%	5.6%	2.5%	-	5.29	committed
F3	25.0%	2.9%	6.9%	5.2%	1.3%	-	5.92	detached

Table 1: Average camera behaviour features with the number of records of each cluster. Clusters C have been extracted from (the records of) the collection areas, clusters F from the fight areas and J from the jump areas. Highlighted in dark grey is the feature related to the main task of the area type.

two stages are composed by a series of small areas, which are categorised according to the game-play experience they offer and the type of challenge they pose to the player. The three categories are *jump*, *fight* or *collection*.

For instance, in the tutorial stage, the first platform is considered a *collection* area, while the second platform, after the first bridge, is considered a *fight* area (see Figure 6a). The set of platforms connecting the third area with the end of the stage are, instead, considered a *jump* area. For the purpose of this study, the areas of the game identify atomic interactive experiences, in which the player is engaged with a single task. Different types of tasks correspond to different game mechanics, hence, different types of interaction. For this reason the behaviour of the camera is analysed separately for the different areas, resulting in multiple models for the same game.

5 Camera Behaviour Models

In this section, we give a brief description of the application of the first phase of the proposed methodology to the game described in the previous section. The content of this section is a summary of the results presented in two articles in which the authors investigated the existence of a relationship between in-game player behaviour and camera control (Picardi et al., 2011) and the possibility to build predictive models of such relationship (Burelli and Yannakakis, 2011).

To build personalised models of camera behaviour, it is necessary to evaluate if different players focus on different object types in the same game situation and if the same player will observe different types of object in different game situations. To test these two hypotheses, we conducted an experiment in which we collected eye gaze, camera and game-play data from participants playing a version of the 3D platformer game described in the previous section.

During the playing session, the participants had control of both the avatar and the camera; the avatar was controlled through the W,A,S and D keys of the keyboard while the camera was controlled using the mouse. The player's actions, the camera movements and the objects observed by the player at each frame were recorded. The estimation of the objects observed was achieved by tracking the

Dataset	Topology	Features	Accuracy
Collection	2, 4	FC, RA, JP	82%
Jump	3, 0	FC, JP	76%
Fight	5, 5	FC, CH, AH, JP	70%

Table 2: Results of the training phase for each dataset: including the topology (in terms of number of neurons in the two hidden layers), the features selected and the accuracy of the model.

players' gaze on the screen and comparing its position with the game objects positions.

After the collection phase, the data was split in segments according to the area category — i.e. fight, collection and jump — and the following features were calculated for each segment: fraction of time spent observing the avatar, fraction of time spent observing fuel canisters, fraction of time spent observing coppers, fraction of time spent observing re-spawn points, fraction of time spent observing platforms, fraction of time spent observing distant objects, and average camera speed. Along with these features describing the camera behaviour we collected the following five features describing the player's behaviour: number of fuel canisters collected (FC), number of times a copper was hit (CH), number of times the avatar was hit by a copper (AH), number of re-spawn points activated (RA), and number of jumps performed (JP).

Twenty-nine persons participated in the experiment generating a dataset of 1168 records corresponding to all segments traversed by the participants with complete gaze information. After a process of removal of incomplete records (resulting in a dataset of 759 records), the records were sorted into three separate datasets according to the segment type. On the obtained datasets, we applied k-means clustering to identify a set of distinct behavioural patterns of the camera across different players. To identify the number of clusters present in the sets, we ran several repetitions of k-means varying the value of k from 2 to 9. The clusters generated at each run were evaluated using a set of five cluster validity indices and the value of k was picked following a majority voting mechanism, resulting in 2 cluster for the collection dataset, 3 clusters for the fight dataset and 3 clusters for the jump dataset[2].

Table 1 shows the average camera behaviour features of each cluster for all three datasets. It is possible to observe that there are certain patterns with respect to which objects gain more focus and how fast is the camera moving on average. In terms of focus, we can identify three types of behaviour: the players almost only focusing on their avatar, the ones focusing mostly on their avatar and the task-related objects, and the ones which keep an even amount of focus on all elements. We introduce three labels to identify these camera behaviours: *self-centred*, *committed* and *detached*. Furthermore, the classification is augmented to differentiate behaviours in which the camera speed was different from the average value (see rightmost column of Table 1).

[2] Further details on the experimental protocol employed to conduct the data collection experiment can be found in Picardi et al. (2011), while the details on how the features are calculated and normalised, and what indices and parameters have been used in the clustering process can be found in Burelli and Yannakakis (2011).

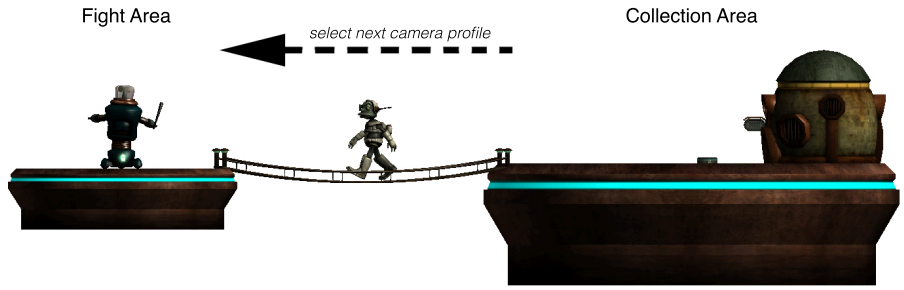


Fig. 7: Example of a transition from one camera profile to another. As soon as the avatar enters the bridge, the neural network relative to the fight area is activated using the player's gameplay features, recorded in all the previously visited fight areas, as its input. The camera profile associated with the behaviour cluster selected by the network is activated until the avatar moves to a new area.

According to the proposed methodology, to employ these camera behaviour patterns for camera behaviour adaptation, we need to investigate the link between the players' in-game behaviour and their way of controlling the camera. For this purpose, we trained an Artificial Neural Networks (ANNs) for each of the three datasets to predict, at point in the game, the camera behaviour type given information about the previous played areas. As presented by Burelli and Yannakakis (2011), three fully connected feed-forward ANNs were trained using resilient back-propagation (Riedmiller and Braun, 1993) on the game-play data (ANN input) from previous segments and the camera behaviour of the current segment (ANN output) using early stopping for over-fitting avoidance. The networks employ the logistic (sigmoid) function at all their neurons.

The initial ANN input features set contained the 5 features describing player behaviour and 3 more values describing the number of fuel canisters (NF), coppers (NC) and re-spawn points (NR) present in the area. Among these features, a subset was selected using sequential forward selection (SFS) (Kittler, 1978) and the best ANN topology was calculated through exhaustive search of all possible combinations of neurons in two hidden layers with a maximum of 30 neurons per layer. As reported by (Burelli and Yannakakis, 2011), each of the selected features is input in the network in two versions: calculated over the last area visited and averaged over the previous ones. The resulting input schemes and the structures of the three ANNs are shown in Table 2; the accuracy in the prediction varies between 70% and 82%.

The number of outputs of each ANN depends on the number of clusters present in the corresponding dataset. Each output signals whether the pattern in input corresponds to that cluster. During the activation of the network, the cluster corresponding to the output with the highest value, is selected.

6 Camera Behaviour Adaptation

The aforementioned models and camera behaviour patterns are the foundations of the camera behaviour adaptation process. The neural networks are used to identify the best camera behaviour pattern to be applied, given the player's previous actions. The camera pattern is then translated into a camera profile that is used to automatically animate the camera.

As shown in Figure 7, the camera behaviour is adapted every time a player enters a new area (or the current area changes its category). At that point, a neural network model is selected and activated depending on the new area category; the output of the network indicates which camera behaviour should be selected from that moment on until the player enters another area. The new behaviour is applied two seconds after the player enters the new area, to avoid multiple repetitive changes if the player moves in and out of the area.

The current area can change its category either if the current area is a *collection* area and the last collectible item is taken or if the area is a *fight* area and the last enemy present is killed. In the first case, a transition between a *collection* and an *empty* area is triggered, while in the second case a transition between a *fight* and a *collection* area is triggered.

When an area changes and the new camera behaviour pattern is selected, this gets converted to a camera profile (Bourne et al., 2008). The combination of these rules is expressed as a weighted linear combination of their satisfaction functions, the resulting function is used by a controller to identify the best camera position at each frame. The controller used in this paper is based on one of the authors previously published work and it combines multiple optimisation algorithms to efficiently place and maintain the camera in an optimal configuration given a certain set of rules (Burelli and Yannakakis, 2010).

Due to the numerical optimisation nature of the controller, the resulting images generated might or might not satisfy completely the camera profile. The satisfaction of the profile depends on whether it is possible to animate the camera to the ideal location in the provided time and whether such ideal location exists. For this reason, the relative weight of each constraint in a profile expresses its priority of satisfaction; this means that, for instance, an object with a low visibility weight will be excluded from the screen before an object with higher visibility weight in case the two constraints conflict.

An example of the latter case can be seen in Figure 8a: the camera is instructed to show the avatar in a certain position on the screen, with a certain angle and size, and the two fuel canisters are required to be in the shot. Given the position of the elements in the virtual environment, there is no camera configuration that can satisfy all the requirements. In this case, the camera controller prioritises the frame constraints with the largest weight, which leads to a shot in which one of the fuel canisters is partially out of screen.

The solver employed in this work supports four type of rules — i.e. frame constraints — that can be used to describe the desired shot: *Object Visibility*, *Object Projection Size*, *Vantage Angle* and *Object Frame Position*. The first frame constraint describes whether an object should be visible or not visible on the screen, the second constraint describes what the size of this object should be, the third constraint defines from which angle the object should be shot and the last one

(a) Two fuel canisters present and the re-spawn point yet to be activated.

(b) One fuel canister present and the re-spawn point already activated.

Fig. 8: Two different viewpoints produced by the same profile: fast committed (C2). The viewpoint changes depending on the number and types of objects available in the area.

describes where the objects should be on the screen. For an in depth description of virtual camera frame constraints the reader is referred to Burelli (2012).

6.1 Camera Profiles

All the profiles corresponding to the different behaviours are built on top of a basic profile. This represents a standard over-the-shoulder shot similar to the viewpoint commonly employed in third-person computer games. It has a camera speed setting of 4.36 (the total average speed of the different clusters) and it is composed by the following frame constraints applied to the avatar:

- *Object Visibility*: the avatar should be fully visible.
- *Object Projection Size*: the projected image of the avatar, either horizontally or vertically, should cover three tenths of the screen.
- *Vantage Angle*: the camera should be placed at an angle relative to the avatar of 170 degrees horizontally and 25 degrees vertically.
- *Object Frame Position*: the avatar should appear at the crossing point of the lower horizontal line and the left vertical line according to the *rule of the thirds* in photography.

This profile serves not only as a basis for the other profiles, but it is used also to drive the camera when the player enters an area that features no jumping, fighting or collection. Each constraint in this default profile is equally weighted (i.e. all weights are equal to 1), meaning that no property is prioritised with respect to the others. The position within the frame has been chosen to balance the image, since the controls are depicted on the right side of the screen, and to grant to the player a good view of the horizon; the default numerical constraint parameters have been selected after a testing phase in which we have searched the values that matched the desired composition principles.

While the four constraints present in the default profile are present in all profiles, the total number of constraints and their weight varies depending on the camera behaviour that needs to be generated. For each profile, all the elements

Cl.	Avatar	Fuel Canisters	Coppers	Re-spawn Points	Platforms	Verti. Angle	Speed
C1	0.59	0.11/nC		0.11/nR		40°	3.3
C2	0.36	0.12/nC		0.07/nR		42°	8.8
J1	0.76	0.46/nC			0.79/nP	0°	2.1
J2	0.74	0.17/nC			0.66/nP	30°	2.7
J3	0.45	0.11/nC			0.56/nP	42°	5.5
F1	0.67		0.09/nE			36°	3.3
F2	0.67		0.48/nE	0.06/nR		40°	5.3
F3	0.25		0.07/nE	0.05/nR		42°	5.9

Table 3: Parameters defining the camera profiles corresponding to each camera behaviour cluster. The first 5 values describe the weight of that specific object category in the composition of the shot, where nC is the number of collectibles present in the area, nE the number of enemies, nR the number of re-spawn points and nP the number of platforms. The sixth value defines the vertical angle at which the camera should frame the scene and the seventh describes the speed of the camera.

that appear observed for more than 5% of the time spent in the corresponding cluster (see Table 1) contribute with one *Object Visibility* constraint. The 5% threshold has been introduced to minimise possible noise in the collected data. The weight of each constraint — i.e. the basic four ones plus the extra visibility constraints — depends on the fraction of time that type of object has been observed in the camera behaviour cluster. If multiple instances of a certain object are present in the area, one *Object Visibility* constraint is assigned to each instance, but the weight of the constraints corresponds to a fraction of the maximum weight inversely proportional to the number of instances.

For example, the camera profile corresponding to the cluster C2 visualised in Figure 8a is comprised of the following frame constraints: the four default constraints with a weight of 0.36, one *Object Visibility* constraint on the re-spawn point with a weight of 0.07 and one *Object Visibility* for each of the two fuel canisters with a weight of 0.6 each. The same frame constraint changes when the number of active re-spawn points and fuel canisters present changes (as shown in Figure 8b). In this case, the profile includes no *Object Visibility* constraints for the re-spawn point and only one *Object Visibility* constraint with a weight of 0.12 for the only fuel canister present.

The fraction of time spent observing distant objects influences the vertical angle value of the *Object View Angle* constraint imposed on the avatar. The vertical angle β equals 45° if the time spent observing far objects equals 0 and it decreases down to 0 as the time increases according to the following formula:

$$\beta = 45^\circ * \frac{(t_{max} - t_{fo})}{t_{max}} \tag{1}$$

where t_{max} is the maximum fraction of time spent observing distant objects in all behaviour clusters (t_{max} is 0.202 in this paper), and t_{fo} is the fraction of time spent observing the far objects in the current cluster. Finally, the dynamic behaviour of the camera is controlled by the average speed parameter of each cluster. The resulting weights and speed values for all the profiles are shown in Table 3.

7 User-Evaluation

To evaluate the efficacy of the adaptation mechanism, we conducted an evaluation using the game described in Section 4. The evaluation has been structured as a within-subject experiment in which each participant plays the same game with two camera control schemes (conditions): (a) the camera is controlled automatically with a constant profile or (b) the camera is controlled by the player profile that is influenced by the player's behaviour. Our experimental hypotheses were: (i) players would prefer to play with camera behaviour that adapts to their in-game behaviour and, (ii) players will achieve better results while playing through the adaptive version of the game.

To test these hypotheses the experiment was designed so that each participant plays both versions of the game (in a randomised order) and, at the end of the second version, she or he expresses her or his preferences between the two experiences.

The participants were initially presented with a short description of the experiment and the game objectives. After this, they were asked to fill-in the following information: age, gender, whether she or he normally plays computer games, whether she or he normally plays 3D platform games and how often in a week (e.g. from 0 to 2 hours per week or from 2 to 5 hours per week). At completion of the questionnaire, each participant started to play the tutorial stage (see Figure 6a). This phase of the experiment was designed to let the player become familiarised with the controls; furthermore, the tutorial had also the purpose to collect an initial set of data about the player's in-game behaviour. During the tutorial, the players had manual control of the camera in order to avoid any learning effect on the subsequent part of the experiment.

After the tutorial, the participants proceeded to the main part of the experiment, in which they played two times through the main game stage (see Figure 6b) once with and once without adaptive camera. The order of the stimuli (camera control paradigm) was distributed to minimise the order effect — i.e. it was swapped at every experimental session — so that about 50% of the participants played the adaptive version of the game first and about 50% the other one.

At the end of the experiment, the participants were asked to express a preference between the two experimental conditions (Game A and Game B) just played and subsequently to provide a motivation for the preference. The preference was expressed through 4-alternative forced choice (4-AFC) questionnaire scheme. The preference questionnaire included four alternative choices: Game A, Game B, Neither or Both Equally. This scheme was chosen over a rating scheme for a number of advantages, including the absence of scaling, personality, and cultural biases as well as the reported lower order and inconsistency effects of ranked-based compared to rating-based self-reporting (Yannakakis and Hallam, 2011).

7.1 Results

Twenty-eight persons participated in the experiment, among which 21 were males, 20 declared to be regularly playing games and 14 declared to be regularly playing three-dimensional platform games. The age of the participants ranged between 18 and 40 years (mean=27.04, standard deviation=4.63). Before the analysis, the

Feature (F)	First (F_1)	Second (F_2)	p-value
Completion time	**172.00**	**148.24**	**0.01**
Canisters collected	7.09	7.78	0.07
Damage inflicted	0.57	0.57	0.50
Damage received	0.30	0.35	0.61
Falls	2.83	2.09	0.20
Jumps	58.87	53.04	0.15
Respawns activated	2.65	2.74	0.29

Table 4: Average performance values of the players with respect to the order of play and test of order of play effect on the features. The p-value is the result of a paired-sample t-test.

data was filtered to remove the logs of the experimental sessions in which the average frame-rate was lower than 30 frames per second. The data about these participants was removed to guarantee that all participants included in the study were exposed to the same interaction experience. After this filtering, the number of participants considered dropped to 23, among which 10 played the first game with adaptive camera control first and 13 played the second game with that condition.

In the reminder of the Section, we first control the effect of the order of play on the participants' performance and on their preferences. Afterwards, we proceed with an analysis of the impact of the adaptive camera behaviours on the participants' preferences and performance.

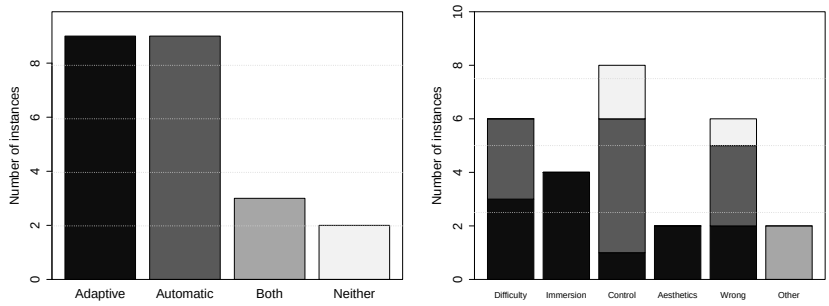
7.2 Order Of Play

To check whether the order of playing the stages affected the participants' performance, we test for the statistical significance of the difference between the features collected in the first stage and in the second stage. Table 4 contains the results of a paired two-sample t-test between each feature collected in the first and in the second game. The statistical test shows that the order of play affects one of the features describing the players' performance: completion time. Moreover, the p-value for the number of collected canisters is very close to the 5% significance level, suggesting the presence of a weak influence of the order of play on this feature.

To test the effect of the order of play on the user's preferences, we follow the procedure suggested by Yannakakis et al. (2008) and we calculate the correlation r_o as follows:

$$r_o = \frac{K - J}{N} \tag{2}$$

where K is the number of times the users prefer the first stage, J is the number of times the users preferred the second stage and N is the number of games. The greater the absolute value of r_o the more the order of play tends to affect the participants' preference; r_o is trinomially-distributed under the null hypothesis. The statistical test shows that no significant order effect emerges from the reported preferences ($r_o = -0.22$, $p\text{-}value = 0.46$).



(a) Preferences (b) Motivations

Fig. 9: Expressed preferences (a) and corresponding motivations (b). The bar colours in the motivations chart describe which preference the motivations have been given for.

7.3 Preference and Performance

Figure 9a displays the total number of selections for each of the four options of the 4-AFC questionnaire. The first noticeable finding is that 18 participants (78.3%) expressed a clear preference between the two camera types. Among these, half of the participants (9 out of 18) stated that they prefer playing the stage in which the camera profile was adapted and the same amount preferred the stage without adaptation. To assess the existence and to evaluate the strength of the relationship between the camera control mechanism and player preference, we model both variables as categorical and binary, we calculate the tetrachoric correlation coefficient (Divgi, 1979) between the two variables and we employ a Pearson's chi-square test of independence to evaluate the significance. The result of such analysis is that there is no significant correlation between player preference and the camera control scheme — i.e. the correlation coefficient is 0.14 with a corresponding p-value of 0.50, which indicates that the adaptive game has not been clearly preferred over the non-adaptive version of the game.

These results confute our first hypothesis that the players would prefer the adaptive camera experience over the non-adaptive one. The motivations given by the participants for their choice (see Figure 9b) reveal that many of the participants who preferred the non-adaptive version of the game over the adaptive one either felt a lack of control or felt the camera was not behaving according to their expectations.

Our second hypothesis is that adaptive camera control will positively affect the player in-game performances. To test this hypothesis we sort the seven features that collectively describe the players' performance during the game into two sets according to the camera control scheme. These are tested to assess whether their means differ significantly.

The first two columns of Table 5 contain the mean values of each feature in the stages featuring the adaptive camera behaviour and the stages without. The last

Feature (F)	Adap. (F_{ad})	Auto. (F_{au})	p
Completion time	156.54	163.70	0.25
Canisters collected	**7.96**	**6.91**	**0.01**
Damage inflicted	0.61	0.52	0.08
Damage received	0.35	0.30	0.61
Falls	**1.83**	**3.09**	**0.05**
Jumps	54.70	57.22	0.54
Respawns activated	2.78	2.61	0.13

Table 5: Average performance values of the players with respect to camera control type. The third column contains the p-values of a paired two-sample t-test between the two measurements across camera control type.

column displays the p-values of a two-sample paired t-test between the two groups of features. These statistics reveal that, overall, the number of collected canisters and the number of falls were affected by the camera control paradigm as these are the only features for which the null hypothesis is rejected. In other words, when playing with adaptive camera, the players were able to collect more items in the same time and with a lower number of errors. This suggests that the adaptation mechanism allowed the players to plan better their exploration and have a better control of the avatar, which supports our second hypothesis.

7.4 Player Types

The results presented so far reveal a positive impact of the adapted camera behaviour on some aspects of players' performance; however, no clear preference for any of the two camera control paradigms is detectable. To achieve a better understanding of the effect of adaptive camera behaviour on the players' experience, we proceed by investigating further whether there was one or more groups with a clear preference and what are their characteristics.

For this reason, we model the participants' preferences as a binary variable, describing whether or not each participant preferred the adaptive camera experience and we correlate this variable with a number of features describing the character-

Feature (F)	r_o	*p-value*
Age	-0.05	0.81
Is a player	0.24	0.26
Is a platform player	-0.16	0.45
Operating system	-0.27	0.21
Average completion time	**0.48**	**0.02**
Average canisters collected	**-0.45**	**0.03**
Average damage inflicted	-0.12	0.57
Average damage received	-0.07	0.75
Average falls	0.34	0.11
Average jumps	0.03	0.87
Average re-spawns activated	**-0.45**	**0.03**

Table 6: Point-biserial correlation between expressed preference and overall player statistics.

Player Type	Players	Time	Canisters	Re-spawn
Experts	10	106.7	8.1	2.8
Intermediates	8	154.0	8.4	3.0
Novices	5	276.8	4.6	1.9

Table 7: Average gameplay statistics and number of players for the different player types identified.

istics of the participants, including their demographic information as well as their overall performance in the experiment — i.e. the average value of their performance in the two conditions played.

As shown in Table 6, this analysis reveals that players with different completion times, canisters collected and re-spawn points activated had different preferences on the camera paradigm. In particular, it appears that players with high completion time, low number of canisters collected and low number of re-spawn points activated had a strong preference for the adaptive camera over the non-adaptive one.

To identify more accurately the groups of participants and their preferences, we cluster them along the three aforementioned variables into three groups using k-means, with the aim of isolating *expert players*, *intermediate players* and *novice players*. Table 7 shows the resulting clusters: the first group (experts) contains 10 players, the second (intermediates) 8, while the last one (novices) only 5. To perform a valid statistical analysis, we merge the intermediate and novice players in one group labelled *non-experts*, as the novice players group is too small for any significant result. On the two resulting groups, we calculate a tetrachoric correlation coefficient r_o between the camera paradigms and the participants' preferences and the results show that there is a significant negative correlation for the expert players — i.e. -0.55 with a p-value of 0.04 — and a positive correlation for the non-expert players — i.e. 0.66 with a p-value of 0.01.

Furthermore, an analysis of the relationship between player performances and camera control paradigm on the players sorted in the two aforementioned groups reveals some interesting results. The results in Table 8 show that, while expert players reported a clearly negative experience with the adaptive camera, there is

Feature (F)	Expert Players			Non-expert Players		
	Adap. (F_{ad})	Auto. (F_{au})	p	Adap. (F_{ad})	Auto. (F_{au})	p
Compl. time	115.07	98.28	0.90	**188.44**	**214.02**	**0.05**
Can. collected	8.20	8.00	0.31	**7.77**	**6.08**	**0.01**
Dam. inflicted	0.70	0.60	0.17	0.54	0.46	0.17
Dam. received	0.40	0.10	0.96	0.31	0.46	0.25
Falls	0.80	0.30	0.84	**2.62**	**5.23**	**0.03**
Jumps	53.20	46.80	0.11	55.85	65.23	0.15
Resp. activated	2.80	2.90	0.70	**2.77**	**2.38**	**0.05**

Table 8: Average performance values of the players with respect to camera control type. The layers are sorted in two performance groups. The third column contains the p-values of a paired two sample t-test between the two measurements across camera control type.

no significant relationship with their performance. On the contrary, for the non-expert players, the adaptation mechanism generated a more appreciated experience and, at the same time, improved their performance significantly — e.g. the number of falls decreases by 51% and the number of collected canisters increases by 28%.

8 Discussion

The results presented so far show that the methodology suggested is partially able to improve the player experience and affect positively the player performance: differences exists between players and between different performance measures. To get a better understanding of the reasons behind such results, in this section, we will reason upon the methodology itself, we will discuss its main contributions and limitations, and we will propose future directions of research to address the limitations.

8.1 Scalability and Generalisability

The implementation and evaluation presented in the article aim both at demonstrating the applicability of the methodology and at evaluating its effectiveness. In terms of applicability, the implementation successfully serves as a proof-of-concept: we managed to get through all the phases described in Section 3 and eventually developed a personalised camera controller for a commercial-grade 3D computer game. Furthermore, as argued in Section 4, the game used in the study can be considered an accurate representative of its genre, allowing us to extend our applicability claim to most third person platformer and action games.

However, a few questions remain unanswered on whether the methodology can be applied during the development of a large production. Aspects such as the cost of the equipment and the time necessary to build the camera behaviour models can hinder the industrial applicability of the methodology. For instance, while the participants to the data collection experiment played an average of 10 minutes and 30 seconds each, the overall collection phase lasted approximately 4 days. This might be a crucial aspect for the applicability of the methodology; however, it is unclear how this aspect would scale for a game that is supposed to be played by hundreds of thousands. Further research would be necessary to find out what is the number of players necessary to get a good coverage of the camera behaviour of a specific target group.

Another aspect that deserves further investigation is the generalisability of the extracted behaviours. More experiments would be necessary to understand whether it is possible to identify camera behaviours that are valid across multiple games and multiple players. By repeating the methodology on different games, it should be possible to understand whether there are common behaviour patterns the span across games.

8.2 Learning and Prediction Algorithms

On top of the quality of the input data, another important contribution to a successful implementation of the methodology is the choice of the learning algorithm

used to model the relationship between the player behaviour and the camera behaviour. While the accuracy of the prediction models employed is relatively high, none of the models used in the evaluation has prediction accuracy higher than 85%. This suggests that there are a substantial number of cases for which the models are unable to predict the right behaviour leading to a selection of incorrect camera profiles during the adaptation phase. Different directions could be pursued to increase the models' accuracy, such as the use of other learning algorithms — e.g. support vector machines — or a global search based feature selection — e.g. a genetic search feature selector. Moreover, a different clustering algorithm could also be employed, as the error might be a consequence of a wrong cluster separation.

8.3 Adaptive Camera Behaviour

The camera behaviour modelling process is based on the concept, developed in one of the authors' earlier work (Burelli, 2013), that camera preferences are directly related to visual composition of the images produced on screen. As a consequence, the aim of the adaptive camera is not to mimic the original camera movements, but to replicate its visual style. The main limitation with this approach is that it is impossible to ensure that the images created by the adaptive camera are visually equivalent to the ones used to build the models for a series of reasons: first, the camera controller has a margin of inaccuracy as it is based on a dynamic optimisation algorithm; second, generating a visually equivalent image might not be possible all the time, as the avatar and the environment change dynamically and, potentially often, to configurations which make it impossible to frame the shot as desired. One example of such condition has been presented in Figure 8a, where the controller cannot, at the same time, place the camera at the desired angle with the avatar while having both fuel canisters completely visible.

This is an intrinsic limitation of automatic camera control in dynamic environments, hence, it is unavoidable in the proposed methodology. As a future work it would be interesting to investigate how accurately this methodology simulates the users' preferences on the camera compared, for instance, with a more direct approach in which the camera behaviour is modelled on the camera position instead of the visualised items. Furthermore, the two approaches should also be compared in terms of user preferences, as an accurate simulation could not necessarily lead to a better user experience.

Another interesting aspect to investigate further is related to the static nature of the user models, which are built *a priori* and do not change during the experience. In future studies, the model itself could be adjusted during the gameplay and tailored to the specific player potentially yielding a more accurate behaviour prediction. The integration of a form of feedback about the user's cognitive state would allow adapting, not only the camera profile, but also the model itself. Such feedback could come from real-time gaze tracking as well as various other modalities, such as computer vision, electroencephalogram (EEG), electromyogram (EMG) and other physiological signals. These also provide a promising way to detect and model the user's affective state (Burelli, 2013), allowing for the better understanding of the relationship between virtual cinematography and the player.

9 Conclusions

This article presented a methodology to generate adaptive virtual camera experiences in computer games based on the authors' previous studies on modelling camera behaviour and its relationship to game-play (Picardi et al., 2011; Burelli and Yannakakis, 2011). An implementation of the methodology was also described to showcase its applicability. Finally, the implementation was evaluated in a user study designed to investigate the effects of the generated camera behaviours on the players.

According to the methodology, the camera behaviour was modelled using a combination of information about players' gaze and virtual camera position collected during a game experiment. The data collected was clustered using k-means to identify relevant behaviours, and the relationship between the clusters and the players' in-game behaviour was modelled using three artificial neural networks. These models of camera behaviour were then used to select — at specific times in the game — a camera profile to provide a personalized virtual cinematographic experience.

A user evaluation of the virtual camera experience generated through this methodology shows that, while not being explicitly preferred by the players, the adaptive camera has a significantly positive impact on the players' performance. Furthermore, by sorting the participants according to their playing skill level, it appears that two groups of players had contradicting preferences over the camera experience: the expert players have a strong preference for the non-adaptive camera experience, while the other players have a strong preference for the adaptive experience.

On the whole, the results show a potential for the proposed methodology to be effectively used to develop games with adaptive visualisation and that such adaptivity can improve the players' appreciation of the game and their performance in the game. For these reasons, we believe that the modelling and adaptation procedure should be investigated further in order to evaluate empirically the generalisability of the methodology as well as to find more robust models and better features to describe camera behaviour.

References

Arijon, D. (1991). *Grammar of the Film Language*. Silman-James Press LA.

Bares, W. H. and Lester, J. C. (1997a). Cinematographic User Models for Automated Realtime Camera Control in Dynamic 3D Environments. In *International Conference on User Modeling*, pages 215–226, Chia Laguna, Italy. Springer-Verlag.

Bares, W. H. and Lester, J. C. (1997b). Realtime Generation of Customized 3D Animated Explanations for Knowledge-Based Learning Environments. In *Conference on Innovative Applications of Artificial Intelligence*, pages 347–354, Providence, Rhode Island, USA.

Bares, W. H., Zettlemoyer, L. S., Rodriguez, D. W., and Lester, J. C. (1998). Task-Sensitive Cinematography Interfaces for Interactive 3D Learning Environments. In *International Conference on Intelligent User Interfaces*, pages 81–88, San Francisco, California, USA. ACM Press.

Bernhard, M., Stavrakis, E., and Wimmer, M. (2010). An empirical pipeline to derive gaze prediction heuristics for 3D action games. *ACM Transactions on Applied Perception*, 8(1):4:1—-4:30.

Blinn, J. (1988). Where Am I? What Am I Looking At? *IEEE Computer Graphics and Applications*, 8(4):76–81.

Bourne, O., Sattar, A., and Goodwin, S. (2008). A Constraint-Based Autonomous 3D Camera System. *Journal of Constraints*, 13(1-2):180–205.

Burelli, P. (2012). *Interactive Virtual Cinematography*. PhD thesis, IT University Of Copenhagen.

Burelli, P. (2013). Virtual Cinematography in Games : Investigating the Impact on Player Experience. In *International Conference On The Foundations Of Digital Games*, pages 134–141, Chania, Greece. Society for the Advancement of the Science of Digital Games.

Burelli, P. and Yannakakis, G. N. (2010). Combining local and global optimisation for virtual camera control. In *IEEE Symposium on Computational Intelligence and Games*, pages 403–410.

Burelli, P. and Yannakakis, G. N. (2011). Towards Adaptive Virtual Camera Control in Computer Games. In Dickmann, L., Volkmann, G., Malaka, R., Boll, S., Krüger, A., and Olivier, P., editors, *International symposium on Smart Graphics*, volume 6815 of *Lecture Notes in Computer Science*, pages 25–36, Bremen, Germany. Springer Berlin Heidelberg.

Christianson, D., Anderson, S., He, L.-w., Salesin, D. H., Weld, D., and Cohen, M. F. (1996). Declarative Camera Control for Automatic Cinematography. In *AAAI*, pages 148–155. AAI.

Christie, M., Olivier, P., and Normand, J. M. (2008). Camera control in computer graphics. In *Computer Graphics Forum*, volume 27, pages 2197–2218.

Divgi, D. R. (1979). Calculation of the tetrachoric correlation coefficient. *Psychometrika*, 44(2):169–172.

Drucker, S. M. and Zeltzer, D. (1994). Intelligent camera control in a virtual environment. In *Graphics Interface*, pages 190–199.

El-Nasr, M. S. and Yan, S. (2006). Visual attention in 3D video games. In *ACM SIGCHI international conference on Advances in computer entertainment technology*, volume 31, page 22, Hollywood, CA, USA.

He, L.-w., Cohen, M. F., and Salesin, D. H. (1996). The virtual cinematographer: a paradigm for automatic real-time camera control and directing. In *ACM SIGGRAPH*, pages 217–224, New Orleans, Louisiana, USA. ACM Press.

Irwin, D. E. (2004). Fixation Location and Fixation Duration as Indices of Cognitive Processing. In *The interface of language, vision, and action: Eye movements and the visual world*, chapter 3, pages 105–133. Psychology Press, New York.

Jhala, A. and Young, R. M. (2005). A discourse planning approach to cinematic camera control for narratives in virtual environments. In *AAAI*, number July, pages 307–312, Pittsburgh, Pennsylvania, USA. AAAI Press.

Kittler, J. (1978). Feature Set Search Algorithms. *Pattern Recognition and Signal Processing*, pages 41–60.

Land, M. F. (2009). Vision, eye movements, and natural behavior. *Visual neuroscience*, 26(1):51–62.

Lino, C. and Christie, M. (2012). Efficient composition for virtual camera control. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 65–70, Lausanne, Switzerland. Eurographics Association.

Mahlmann, T., Drachen, A., Togelius, J., Canossa, A., and Yannakakis, G. N. (2010). Predicting player behavior in Tomb Raider: Underworld. In *IEEE Conference on Computational Intelligence and Games*, pages 178–185, Copenhagen, Denmark.

Phillips, C. B., Badler, N. I., and Granieri, J. (1992). Automatic viewing control for 3D direct manipulation. In *ACM SIGGRAPH Symposium on Interactive 3D graphics*, pages 71–74, Cambridge, Massachusetts, USA. ACM Press.

Picardi, A., Burelli, P., and Yannakakis, G. N. (2011). Modelling virtual camera behaviour through player gaze. In *International Conference on Foundations of Digital Games*, pages 107–114, Bordeaux, France. ACM Press.

Pinelle, D., Wong, N., and Stach, T. (2008). Heuristic evaluation for games: usability principles for video game design. In *ACM CHI*, CHI '08, pages 1453–1462, Florence. ACM Press.

Pontriagin, L. S. (1962). *Mathematical Theory of Optimal Processes*. Interscience Publishers.

Ranon, R. and Urli, T. (2014). Improving the Efficiency of Viewpoint Composition. *IEEE Transactions on Visualization and Computer Graphics*, 2626(c):1–1.

Riedmiller, M. and Braun, H. (1993). *A direct adaptive method for faster backpropagation learning: the RPROP algorithm*. IEEE.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation. *Nature*, 323:533–536.

Sundstedt, V., Stavrakis, E., Wimmer, M., and Reinhard, E. (2008). A psychophysical study of fixation behavior in a computer game. In *Symposium on Applied perception in graphics and visualization*, pages 43–50, Los Angeles, California, USA. ACM.

Togelius, J., Kastbjerg, E., Schedl, D., and Yannakakis, G. N. (2011). What is Procedural Content Generation ? Mario on the borderline. In *International Workshop on Procedural Content Generation in Games*, page 6, Bordeaux, France.

Tomlinson, B., Blumberg, B., and Nain, D. (2000). Expressive autonomous cinematography for interactive virtual environments. In *International Conference on Autonomous Agents*, page 317, Barcelona, Spain.

Ware, C. and Osborne, S. (1990). Exploration and virtual camera control in virtual three dimensional environments. *ACM SIGGRAPH*, 24(2):175–183.

Wolf, M. J. P. (2001). Genre and the video game. In Wolf, M. J. P., editor, *The medium of the video game*, chapter 6, pages 113–134. University of Texas Press.

Yannakakis, G. N. and Hallam, J. (2011). Rating vs. Preference: A comparative study of self-reporting. In *Affective Computing and Intelligent Interaction Conference*, pages 437–446, Memphis, TN, USA. Springer-Verlag.

Yannakakis, G. N., Hallam, J., and Lund, H. H. (2008). Entertainment capture through heart rate activity in physical interactive playgrounds. *User Modeling and User-Adapted Interaction*, 18(1-2):207–243.

Yannakakis, G. N., Martínez, H. P., and Jhala, A. (2010). Towards affective camera control in games. *User Modelling and User-Adapted Interaction*, 20:313–340.

Yarbus, A. L. (1967). *Eye movements and vision*. Plenum press.

(1)Dr. Paolo Burelli

Dept. of Architecture, Design and Media Technology, Aalborg University Copenhagen, AC Meyerænge 15, 2450 Copenhagen, Denmark

Dr. Paolo Burelli is an Assistant Professor at Aalborg University Copenhagen in the Department of Architecture, Design and Media Technology, his research work focuses on intelligent user interfaces in computer games. He received a Ph.D. in Artificial Intelligence from the IT University Of Copenhagen in 2012, and a Master in Computer Science from the University Of Udine in 2007. Paolo has published articles on virtual cinematography, human-computer interaction and artificial intelligence and he is a member of the IEEE Computational Intelligence Society Games Technical Committee. His research interests include computer games, human-computer interaction, perception and artificial intelligence with a particular focus on implicit interaction and virtual cinematography.

(2)Professor Georgios N. Yannakakis

Institute of Digital Games, University Of Malta, Msida, 2080, Malta

Georgios N. Yannakakis is an Associate Professor at the University of Malta (UoM), Institute of Digital Games. He received the Ph.D. degree in Informatics from the University of Edinburgh in 2005. Prior to joining the Department of Digital Games, UoM, in 2012 he was an Associate Professor at (and still being affiliated with) the Center for Computer Games Research at the IT University of Copenhagen. He does research at the crossroads of AI (computational intelligence, preference learning), affective computing (emotion detection, emotion annotation), advanced game technology (player experience modelling, procedural content generation, personalisation) and human-computer interaction (multimodal interaction, psychophysiology, user modelling). He has published over 150 journal and international conference papers in the aforementioned fields. He is an Associate Editor of the IEEE Transactions on Affective Computing and the IEEE Transactions on Computational Intelligence and AI in Games.