Analysis of Third Person Cameras in Current Generation Action Games

Jonathan Schramm

Degree Project in Game Design, 15 ECTS Credits
Game Design and Graphics, spring 2013
Supervisors: Masaki Hayashi, Leo Sandberg
Examiner: Steven Bachelder
Abstract

The purpose of this project was to research the virtual camera systems used in current generation third person action games and to see what could be improved upon. To do this, different camera shots were categorized into camera archetypes, which also include post process and lens effects used. Information about the games was acquired by either looking through the game’s settings or by observing gameplay. Finally the results were compared to each other as well as the film industry and several improvements regarding usage of different features and camera shots were suggested.

Keywords: Third person, virtual camera, camera system, action games, analysis.
## Contents

1. Introduction ................................................................................................................................. 4
2. Camera Features and Effects ........................................................................................................ 4
3. Method ........................................................................................................................................... 6
   3.1 Criteria for Choosing Games .................................................................................................... 6
   3.2 Graphic Settings and Configuration Files ................................................................................ 6
   3.3 Observations ........................................................................................................................... 8
4. Observations of the games ............................................................................................................. 9
   4.1 Assassin’s Creed III .................................................................................................................. 9
   4.2 Batman Arkham City ............................................................................................................... 10
   4.3 Dark Souls ............................................................................................................................ 11
   4.4 Sleeping Dogs ....................................................................................................................... 11
   4.5 Tera Online ........................................................................................................................... 12
   4.6 Tomb Raider .......................................................................................................................... 14
5. Analysis and Discussion ................................................................................................................. 15
   5.1 Camera archetypes .................................................................................................................. 15
   5.2 Comparison of each game studied .......................................................................................... 21
   5.3 Camera Features Discussion .................................................................................................... 22
   5.4 Camera Movements Discussion ............................................................................................... 29
   5.5 General Discussion .................................................................................................................. 32
6. Conclusion ..................................................................................................................................... 33

References .......................................................................................................................................... 34

Appendix ............................................................................................................................................ 36
1 Introduction

This paper investigates the cameras currently used in third person games, their usage and their features. What the designers decided to implement during gameplay and what they left out or possibly used in cut scenes instead.

There is not a lot of information on virtual camera systems in games as opposed to the extensive work done in the film industry, and especially not in the category of third person cameras in action games. While the game industry can try to mimic the camera techniques used in movies there is still a difference in non-interactive and interactive media which needs to be taken into consideration according to Georg Backer (Backer 2011).

Another aspect of the cameras is different forms of lens effects and post processing effects. In games these may try to emulate reality (lens flares, bloom) and in some cases they are more stylized (overuse of motion blur). When a movie needs to a focal length for a specific shot the technology is already there. But if you don’t have the feature in the game engine you are working with, that is a problem which has to be solved.

The purpose of this study is to create an overview of the AAA game cameras and to see what areas can be improved upon, be it technical features or camera movement.

2 Camera Features and Effects

Most of the effects used in games are based on phenomena which are either perceived by our eyes or by a camera lens. The following are some of the more commonly used effects and features. Some of them are used by every game but differently (Degrees of freedom) while others are used by some games and not at all by others (Lens flare). Some post processing effects has also been left out such as anti-aliasing as it is very old and expected by every new AAA game released.

Bloom

As defined by Epic Games’ article on bloom post processing effects (Epic Games 2012):

Bloom is a real world light phenomena that greatly can add to the perceived realism of a rendered image at a moderate render performance cost. Bloom can be seen by the naked eye when looking at very bright objects that are on a much darker background. Even brighter objects also cause other effects (streaks, lens flares) but those are not covered by the classic bloom effect.

Camera shake

This is perhaps one of the most basic effects. It is created by simple twitching camera movements while not letting any individual twitch take the camera too far away from its initial position. It is described in more detail by Muir and John Kenneth (Muir and John Kenneth 2004).

Degrees of freedom

According to Weisstein (Weisstein 2007) the degrees of freedom is defined as ‘A rigid body in three-dimensional space has six degrees of freedom: three linear coordinates for defining the position of its center of mass—or any other its point—and another three Euler angles defining relative rotation around the body's center of mass’. For example first person shooter games usually have 5 degrees of freedom as the avatar is capable of moving in two directions on the ground, can jump or crouch and look around by yawing or pitching.
Distortion
This effect can be thought of as a normal map which is applied to the entire picture after the scene has been rendered. More information about normal maps can be found in Epic Games’ article regarding materials (Epic Games 2012). Although it only works on the X and Y axes, as horizontal and vertical. The result is a scene with distorted UV coordinates depending on the map or values used according to Epic Games’ article on distorting post processing effects (Epic Games 2012).

Depth of Field
This is an effect which emulates the behavior of real camera lenses. The effect focuses on a distance and allows the objects around that distance to stay sharpened while objects further away or closer to the lens becomes blurred, Shene (Shene 2004).

Field of view
Field of view defines how much the camera can see, what is observable for a camera, Atkins (Atkins 2004), usually defined in degrees. In reality this is changed by changing the lens and by doing that, the focal length.

Lens flare
Lens flare is a phenomena created by bright light sources scattering light in lens systems in a way which was not intended according to Shene (Shene 2000). This effect has been created in game engines as well to create a form of realism.

Motion-blur
This effect causes blurs by emulating objects moving while the camera shutter is still open, or moving the camera itself while the shutter is still open, according to Potemsil and Chakravarty (Potemsil and Chakravarty 1983).

Radial blur
While this effect is the same as regular motion blur in the film industry the technique is not exactly the same in games. According to Epic Games’ article on radial blur (Epic Games 2012):

The radial blur effect allows you to mimic the look of a shockwave coming from an explosion. The effect blurs the screen in the direction emanating from the center of the radial blur actor that is placed in the scene.

A similar effect can also be achieved if the camera moves fast enough forward or backward.

Screen space ambient occlusion
Screen space ambient occlusion is an algorithm which calculates ambient occlusion in real time (Mittring 2007). While there are other real time ambient occlusion algorithms, this one is the one mainly used for the games studied. Other ways of creating ambient occlusion in real time includes temporal screen space ambient occlusion created by Mattausch et al. (Mattausch, Scherzer and Wimmer 2009) and ambient occlusion volumes created by M. McGuire (M. McGuire 2010).
3 Method

3.1 Criteria for Choosing Games

To determine what games were to be studied a short list of required criteria was compiled. The purpose was to filter out what games have a high enough standard to be considered current generation so that the features of the game are up to date with the game industry. This is the list used:

1. The game must be in third person.
2. The player should only be controlling one avatar at a time.
3. The genre of the game should be some form of action game.
4. The game should not have been released more than 3 years ago.
5. It should be an AAA title.

The definition of AAA title is debatable but is generally considered to be a game made by a large studio with a large budget, having high-quality graphics and being marketed and sold to a broad market, Orland, Steinberg and Thomas (Orland, Steinberg and Thomas 2007). Additionally all the games presented in this study have been analyzed with a computer as platform to make the process simpler. Using this list the following games were chosen:

Assassin’s Creed III. This game was released by Ubisoft in 2012 and is an action-adventure game.

Tera Online. Also known as TERA. This game is an MMO and was released by Bluehole Studio and was released in 2012 in North America and Europe.

Tomb Raider. An action-adventure game developed by Crystal Dynamics and released in 2013. Note that there are other video games with the exact name but only one of them was released in 2013.

Dark Souls. This game was developed by From Software and was released on PC in 2012. It is considered an action RPG according to IGN (IGN 2011).

Batman Arkham City. This game is an action-adventure game released by Rocksteady Studios in 2011 for PC.

Sleeping Dogs. An action-adventure game which was developed by United Front Games together with Square Enix London according to Nutt (2011), and released in 2012.

3.2 Graphic Settings and Configuration Files

By identifying the different options in a game’s graphics menu (Figure 1) it is possible to deduct some of the features of the camera. These options can include for example motion blur and depth of field or hint about other different post-processing or lens effects.
Figure 1: the graphic settings menu of Tomb Raider.

Another source of information is the game’s configuration files (Figure 2). They can normally be found somewhere in the game’s directory as initialization files where different parameters and their values are listed as plain text according to Microsoft (Microsoft 2008). There are other types of configuration files and the initialization file format is only for the Windows operating system. Here it is possible to find options which are not available in the graphic settings menu.
3.3 Observations

Observation is done simply looking at the game played in real-time. This involves watching videos of the game being played, watching another person play, or playing yourself. By doing this it is possible to find camera features which are listed neither in the graphic settings menu nor in the configuration files (Figure 3). The most important thing however is that it is possible to observe the camera’s movement, which is not determined by a single parameter.
4 Observations of the games

4.1 Assassin’s Creed III

(1) Graphic Settings and Configuration Files

The menu does not allow for much insight into the game’s camera. Only options such as texture quality and resolution are present. There are only other post processing effects in the menu which we can take for granted by now, such as anti-aliasing.

Assassin’s Creed III for the PC does have a configuration file but there are no relevant options here which are not listed in the graphic settings menu. This file can normally be found in the Assassin’s Creed III directory located in Documents.

(2) Observations

Observed features:

- Bloom
- Motion blur
- Radial blur
- Degrees of freedom: 5

In the most common state for the avatar, standing or walking/running, the camera is a medium shot from behind and slightly above. It is possible for the player to orbit the camera around the avatar to create other angles when standing still. If the camera is moved horizontally while the avatar is moving the forward direction will change accordingly to the camera. So while it is possible to have a moving shot from the side this also means that the player has to control the avatar to move sideways over the screen. Vertically, the camera’s angle does not affect the avatar and the player can create anything from a low to a high angle shot.

The distance of the camera mostly remains the same while the avatar is in one of these common states although it does slightly back up the faster the avatar is moving. A few game moves does change the camera’s position more, such as zooming out further and locking in place when the player performs a leap of faith (jumping down from a high place) or locking it to an over the shoulder shot as the player aims a flintlock pistol.

As the player enters a new city, a non-interactive aerial establishing shot will be shown. These shots are also used for example when the player chooses to scan an area from a very high place such as a tower (it’s called synchronization in game). Some events also trigger these kinds of shots but they’re not always aerial. Every once in a while an extreme wide shot is taken from the ground as well.
When fighting enemies the player can perform finishing moves and counter attacks. These moves generally lock the camera in place or force it to move in a specific way, but sometimes they act more interestingly. A few of these moves changes the camera’s position more drastically, zooming in to a close-up or over the shoulder shot as the animation is played. Some of these moves even make the camera cut to a new position. Others place the camera at a specific angle but give the player control of the rotation of the camera, but if the player lets go of the controls it will revert back to its initial rotation.

While the player is controlling the camera, the avatar will stay very close to the middle of the screen. But if the player is performing a move where the camera is controlled by a script instead, the character rarely remains close to the middle. The player is also capable of a special action called eagle vision which desaturates the screen and adds stronger colors to specific objects with which the player can interact.

**4.2 Batman Arkham City**

(1) Graphic Settings Menu and Configuration Files

To access the game’s graphic settings menu we have to run a file named BmLauncher.exe within the game’s directory, unless it’s the first time the game is started in which case it will run automatically. Here we find options to configure motion-blur, distortion, lens flare and some form of real time ambient occlusion.

The game’s configuration files makes it evident that game also uses depth of field and bloom although to what extent is not specified. It is also worth noting that the game has a lot of configuration files spread throughout the entire directory but the one handling in game graphics is located at Batman Arkham City/Engine/Config/BaseEngine.ini. Many of the options found in the graphic settings menu can be found here once again.

It is also possible to edit another initialization file to change the field of view value by binding a value to a hotkey but this is most likely not an intended feature as it requires adding completely new information to the file. Also, it does not give any clues about the default field of view value.

(2) Observations

Observed features:

- Camera shake
- Distortion
- Degrees of freedom: 5

The camera varies depending on the state of the game but in general it will take a wide shot from a high angle. While the player can arc it around the avatar as desired, it will also move around on its own during combat to show attacks from different angles.
There are also several other states where the camera works differently, such as flying which causes the camera to zoom out to a wider shot but still going closer to a medium shot if the player loses speed during flight. Aiming many of the different weapons or standing at certain vantage points will make the camera go to an over the shoulder shot instead. Some weapons are also capable of aiming from a point of view shot, or first person view. Another occasion where the camera will change position is when the player performs finishing moves. If the player goes indoors the camera will go from the default wide shot to a medium shot from the waist up instead.

The player can also target a specific enemy which will lock the camera behind the avatars back while allowing the avatar to arc around the enemy instead of moving normally. Another useful command for the player is to enter detective vision which colors the lens blue and reveals previously unknown information such as the location of hiding enemies by giving some objects another color or via a graphical user interface.

4.3 Dark Souls

(1) Graphic Settings Menu and Configuration Files

Motion blur is the only point of interest in the graphic settings menu with a simple on/off button. There is also the option of anti-aliasing which does count as a post processing effect but is not relevant. The menu is very sparse and there is no configuration file for the game either.

(2) Observations

Observed Features:

- Bloom
- Camera shake
- Degrees of freedom: 5

As previously this game also utilizes a camera which can arc around the character both horizontally and vertically. The direction of the camera changes the direction of forward for the controls as well but not the abilities of the avatar unless it moves or rotates. By default the camera is taking a wide shot directly from behind the avatar and slightly from above, this is where the camera will arc to if the player presses the reset camera button.

The player can also aim with a bow which causes the camera to take an over the shoulder shot for locking on to a target. Aiming also changes the camera position but it’s too far forward to be considered an over the shoulder shot. Locking a target means that the camera will be locked behind the avatar which is now arcing around the enemy in the same way that the camera usually arcs around the avatar. Obviously the avatar cannot leave the ground as the camera can levitate, making up and down give the avatar movement forward or backward instead.

4.4 Sleeping Dogs

(1) Graphic Settings Menu and Configuration Files
By opening the Advanced Options tab of the graphics settings menu screen space ambient occlusion and motion blur can be configured to an extent. No parameters display their actual value but instead the user can choose from preset options such as off, normal and high or by pulling a slider.

While the game does use configuration files none of them are related to any graphic settings which are not possible to find in the game.

(2) Observations

Observed features:

- Bloom
- Camera shake
- Degrees of freedom: 5

As in so many previous examples, the camera is capable of arcing around the avatar at the player’s command. The camera is usually taking a wide shot from a high angle, from behind the character unless changed by the player. If it is rotated horizontally into a direction which is not forward for the avatar, the avatar will change direction. However if it is rotated vertically instead, the camera will arc back over a period of time. In short, rotating the camera horizontally affects the avatar afterwards while rotating it vertically affects the camera afterwards.

Many player actions also change the camera. Talking to certain shopkeepers for example will cause the camera to zoom in for a two shot and aiming guns will change it to an over the shoulder shot. During some of these game states the camera may lock its position while still allowing the player to change the rotation to an extent. The player can also check the avatar’s cell phone which changes the camera to take a medium shot directly from behind. In contrast to many other games the targeting system in this game doesn’t lock the camera position behind the avatar, only the avatar’s movement.

During combat there are many different finishing moves and combos which can change the camera in different ways such as going into a close-up shot and sometimes even closer. Other times it will simply arc to another angle while still taking a wide high angle shot from above, possibly to show what is happening around the avatar better. Either way the player almost always retains some control over the camera.

There are also some parts of the game where the camera follows a script instead. For example the player might enter an area which works as an arena and the only way out is to fight a number of enemies. In some of these instances the camera will take a shot from an even higher angle and instead of having a position relative to the avatar it might always keep a certain height, looking down on the combat.

4.5 Tera Online

(1) Graphic Settings Menu and Configuration Files

The graphic settings menu does not provide any direct evidence of existing camera features but rather there are sliders with more abstract names such as lightning enrichment, which could mean there is a bloom effect or maybe lens flares.
However there is a configuration file directory located under /TERA/Client/S1Game/Config for the default installation which provides more information as shown in figure 4. Looking in the DefaultEngine.ini file it becomes clear that the game has a default field of view value of 60 but it is also configurable. There is also a file named S1Engine.ini which tells us that bloom, motion blur, depth of field, distortion, lens flares and some form of real time ambient occlusion post processing effect exists within the game.

![Figure 4: part of a configuration files used by Tera Online.](image)

Note that the file is very similar to the Batman Arkham City configuration file shown before. The games are using the same engine.

(2) Observations

Observed features:

- Camera shake
- Degrees of freedom: 5

In this game the player has several ways of controlling the camera, even though it’s always in relation to the avatar. The default is a wide-shot from behind and above, and slightly to the side of the avatar and the player may arc the camera around as in previous games. The player can also change the camera by moving it sideways or pedestaling up or down. It is also possible to zoom in creating medium or close-up shots. If zooming in completely the camera will change to a point of view shot, also called first person view.
The camera is always stuck in position relative to the avatar and will always move at the exact same velocity and angle as the avatar, unless impossible due to colliding with terrain. The rotation however will not change as the avatar’s rotation changes. Instead it will stay the same and only change when the player commands. The player cannot center the camera completely on the avatar either; it’s always going to be slightly to the right or left.

These are how the player mainly interacts with the camera but there are also a few other ways. The player may choose at any point in time to change the camera back to its default position and rotation. Also, some game moves affect the camera. For example teleporting the avatar causes the camera to cut and be placed behind the avatar in the new position and rotation.

Another interesting feature of the game is when the player flies into one of the large regions which the world is split into. While the camera is still fixated on the avatar while still allowing control, the avatar is forced to follow a specific path in the air. This results in an aerial establishing shot which shows the player the region that they are soon going to land in and still lets the player have control of the camera.

### 4.6 Tomb Raider

(1) Graphic Settings Menu and Configuration Files

By opening the graphic settings menu, which is available before the game is launched, you can see that the game uses a configurable depth of field value and screen space ambient occlusion. There is also an option named Post Processing but the name is not enough as there is no explanation of it. The in-game menu is the same but with additional user interface graphics.

To open up the configuration for this game it is required to have a look at the system registry. This can be done by running regedit.exe and locating HKEY_CURRENT_USER\Software\Crystal Dynamics\Tomb Raider\Graphics. Here it is possible to edit the monitor height, width and aspect ratio values which indirectly then affects the field of view. There’s more information on the registry in Microsoft’s article (Microsoft 2012).

(2) Observations

Observed features:

- Bloom
- Motion blur
- Lens flare
- Distortion
- Camera shake
- Degrees of freedom: 6
In general the camera will take a medium close-up shot from the waist up and behind while the avatar is standing or running. If the player is outdoors the camera will change to more of a wide shot. The player is able to arc the camera around the avatar both horizontally and vertically which changes the controls and the directions. The camera shake or head-bob motion also increase with the velocity of the avatar, and it also involves rotation on the z-axis. Other common moves such as aiming a weapon will lock the camera to an over the shoulder shot instead.

The camera is also heavily affected by scripts, activated by either changing the state of the avatar or encountering specific events. Examples of avatar states that changes the camera are when the player decides to climb a rope and the camera backs off to a wide shot or if the player performs a silent kill and the camera cuts to a medium shot from the front and arcs back to its initial position behind the player while the animation plays. As for the events these can include non-interactive wide establishing shots as the player enters an area or changing into high angle medium shot while still allowing the player to control some of the rotation.

Something worth noting for the camera is that even though it enters states often which are scripted for a specific event or state, the player often retains an amount of control over the camera.

There are also plenty of other commands that affect the camera indirectly or directly. The player can for example activate survival instincts which is a post processing effect which simply desaturates most of the screen, swap shoulder of the over the shoulder shot when aiming, zoom while aiming or using items such as the rope ascender which zooms in or out depending on the situation.

In combat there are also several special attacks for the different weapons which enter a small cut scene when used. These are mostly scripted so that the camera cuts and shows the action from a specific angle but some of them do not change the camera and often the player can still arc it around the avatar as usual.

5 Analysis and Discussion

5.1 Camera archetypes

There are several patterns which are observable throughout the different games which have been studied in this paper. If more games should be added other camera archetypes might become obvious as well. This is also a snapshot in time of games made in the years 2011-2013 so will change over time, especially with new technology being released such as Unreal Engine 4, Epic Games (Epic Games 2012), or PlayStation 4, Sony Computer Entertainment Europe (Sony Computer Entertainment Europe 2013).

Many of the patterns are following the film industry’s standard shots. Over-the-shoulder shots and aerial establishing shots are nothing new. However there is a need to convert these techniques depending on the game industry’s needs and apply them while also accounting for the interactivity of games.
Aerial establishing shot
By reaching specific areas in the game the player can take a look at the environment by having the camera move back for an extreme wide-shot and then pan around following a script as shown in figure 5. This is activated if the player reaches a certain point (cut scene) or by reaching a certain point and pressing a button (game move).

![Figure 5: an establishing shot as the player flies into an area in Tera Online.](image)

Arcing default camera around the avatar at a fixed distance
The camera can arc around the avatar (Figure 6) both horizontally and vertically, similar to an orbit. However should a piece of terrain or larger object block the view a technique will be needed to reveal the avatar again, such as going closer or making the blocking object transparent. Controlling the camera requires either four buttons (left, right, up, down) or a device capable of detecting 2-dimensional movement such as a mouse or an analog stick.
Camera distance tied to avatar position
The camera zooms in or out depending on where the avatar is located. It might zoom out when the character is standing on a high ground like in figure 7 or zoom in if it is located further down than usual.

Figure 7: the camera is further back as the avatar is on an elevated position in Batman Arkham City

Camera distance tied to avatar velocity
Depending on the speed at which the avatar travels the camera moves closer or further away from the avatar. This is important to create lead room and to allow the player to see better when they are moving at an increased speed (Figure 8).
Forced camera movement

Sometimes the game might forcibly move the camera to a specific position and rotation before allowing the player to take control of the camera again. This is used mainly to show the player what the designer wants them to see, such as the way forward or an important object with which they will need to interact with to progress. While this is similar to both cut scenes and a position forced by a game move, this category is neither activated by the player directly nor does it take control of the avatar away from the player. The player is still able to control the avatar even though the camera is controlled completely by scripts.

Locked over-the-shoulder shot targeting systems

Many games have targeting systems which allows the player to lock on to one enemy (Figure 9). This locks the camera for an over the shoulder shot at a medium to wide shot range. The camera controls are either rendered useless or now used to command the avatar instead. For the player to lock on to an enemy they either press down or hold a button or key. This is not to be confused with the over-the-shoulder shot when the player is aiming as the avatar moves and acts differently depending on the game state and which game it is.
Figure 9: targeting an enemy in Dark Souls.

**Over-the-shoulder shot while aiming**
The camera zooms in or otherwise changes its position to a shot from over-the-shoulder of the player controlled avatar as in figure 10. This game state is somehow activated by the player either by holding or pressing a specific button. Most commonly it is used with firearms or other forms of ranged weapons but it can also be used with tools such as the rope ascender in Tomb Raider.

Figure 10: an over-the-shoulder shot in Tomb Raider.
Partial camera control

During some animations activated with game moves or during cut scenes, the game will still allow the player to control the camera to an extent. While the player might not move or rotate the camera to look at something completely different than the designer intended they may affect the way it is portrayed. This technique is not used by itself but rather it is combined with others. For example the player might perform a game move which puts the character in a longer animation and cuts the camera to another position but the player is still allowed to control the camera to a degree.

Player controlled post-processing effects

The player can activate or deactivate post-processing effects which may or may not affect the game state. Post-processing effects not affecting the game state only lets the player see more clearly by highlighting objects such as enemies or power ups (Figure 11). It could also reveal previously unknown information and let the player perform new game moves which they could not in the initial game state. This would be an action which also affects the game state.

Figure 11: a player controlled post processing effect showing civilians and enemies in Assassin’s Creed 3.

Relative positions and angles for game moves

When the player performs certain game moves she will lose control of the camera. The camera will show what is happening from a position and angle calculated by the game which is relative to the event (Figure 12). The camera can either move there over time or cut to get there instantly. Some games still allow the player to control the camera to an extent. For example they might be allowed to rotate the camera a few degrees in any direction but not enough to turn away from the subject (partial camera control archetype).
5.2 Comparison of each game studied

The following tables show the types of camera movements used and which technology. Figure 13 and 14 shows the technological differences while figure 15 and 16 shows the camera movement used. Some of the categories are more common: Lens flare, motion blur, screen space ambient occlusion, bloom, camera-shake, arcing default camera, over-the-shoulder shot while aiming and partial camera control all have 4 or more games using it. There are also a few more things which are very common; the field of view is mostly unknown, the depth of field is unknown in every game and most game cameras have 5 degrees of freedom.

<table>
<thead>
<tr>
<th></th>
<th>Field of View</th>
<th>Lens Flare</th>
<th>Motion Blur</th>
<th>Radial Blur</th>
<th>Screen Space Ambient Occlusion</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Assassin's Creed 3</strong></td>
<td>Unknown</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Batman Arkham City</strong></td>
<td>Unknown</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Dark Souls</strong></td>
<td>Unknown</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>Sleeping Dogs</strong></td>
<td>Unknown</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Tera Online</strong></td>
<td>60</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Tomb Raider</strong></td>
<td>Unknown</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

*Figure 13: technical comparison of the studied games. Part 1.*

<table>
<thead>
<tr>
<th></th>
<th>Bloom</th>
<th>Camera Shake</th>
<th>Degrees of Freedom</th>
<th>Distortion</th>
<th>Depth of Field</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Assassin's Creed 3</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>5</td>
<td>No</td>
<td>Unknown</td>
</tr>
<tr>
<td><strong>Batman Arkham City</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>5</td>
<td>Yes</td>
<td>Unknown</td>
</tr>
<tr>
<td><strong>Dark Souls</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>5</td>
<td>No</td>
<td>Unknown</td>
</tr>
<tr>
<td><strong>Sleeping Dogs</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>5</td>
<td>No</td>
<td>Unknown</td>
</tr>
<tr>
<td><strong>Tera Online</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>5</td>
<td>No</td>
<td>Unknown</td>
</tr>
<tr>
<td><strong>Tomb Raider</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>6</td>
<td>Yes</td>
<td>Unknown</td>
</tr>
</tbody>
</table>

*Figure 14: technical comparison of the studied games. Part 2.*
### 5.3 Camera Features Discussion

**Bloom**

In the studied games we can see it is heavily used and most light sources uses a bloom effect to some degree (Figure 17). Since the render performance cost is considered moderate, overusing this post-processing effect is not possible for a functioning game.

![Figure 17: the neon signs have a noticeable bloom effect in Sleeping Dogs.](image)
Every game studied in this paper uses the bloom post-processing effect to some degree. Using it more or less than reality should be considered stylizing the game since every game engine used for these games is capable of creating a bloom effect. The exception would be technological limitations of the target platform or optimization.

**Camera shake**

In the games studied camera shake is used mostly in conjunction with larger environmental events like Tomb Raider’s earthquakes and explosions or during combat like in Sleeping Dogs when the avatar punches or gets punched by an enemy. This is not a very advanced feature but even so every game uses it for different purposes.

Some games like Tomb Raider, and sometimes Batman Arkham City and Sleeping Dogs as well, use the effect heavily. One might argue that this is a stylistic choice, that driving a car at high speeds in Sleeping Dogs justifies a constant camera shake. The problem is that this can affect the user physically and make them sick according to Ebert (Ebert 2007). So while the designer is free to use camera shake for their purpose, it sometimes needs to be toned down or at least it should not be used for extended periods of time.

**Degrees of freedom**

Regarding the degrees of freedom the games are very similar. Movement on all axes is possible as well as pitching and yawing. What is more uncommon, at least during gameplay, is rolling. This kind of movement is only seen in Tomb Raider, as in figure 18, out of all the games in this paper.

![Figure 18: The camera rolls during gameplay in Tomb Raider at the Solarii Fortress level.](image)

However many of the game engines used for these games are capable of rolling the camera systems. While they do not use it for gameplay it is used in cut scenes instead. This is true for both Batman Arkham City and Sleeping Dogs. As this technique adds a new dynamic to the camera system designers would be wise not to dismiss it. For example Reed (Reed 1972) points out that it can be used to make the viewer feel more uncomfortable.
**Distortion**

The games which used this effect are Batman Arkham City, Tera Online and Tomb Raider. Distortion is often used for water on the lens, the heat of a fire distorting what lies behind and in the case of Tera Online, supernatural phenomena.

*Figure 19: water on the lens distorts the picture in Tomb Raider.*

The effect might have been too advanced for the other game engines which results in it not being used. As seen in figure 19 it is a very powerful effect capable of distorting basically everything. With time we should be seeing this effect in more games when technology advances.

**Depth of field**

Since the games studied are almost always focusing on the avatar, only with a few exceptions, objects are mostly more blurred the further away they are as in figure 20. It is not often that an object is allowed to appear between the avatar and camera, as the camera script in all of these games will try to change position to once again show the avatar completely. But when objects are in fact closer to the lens they are not noticeably blurred.

This is a simplification of reality and camera lenses. When an object does appear in between the game camera and the avatar, such as a shrubbery without a hitbox, it should be blurred. If the depth of field is implemented in such a way that the avatar is always in focus it would make sense for objects closer to the lens to be blurred.
Assassin’s Creed 3 for example does not use any heavy depth of field effects while the player is playing the game but when a cut scene is played it is very noticeable. Of course the game does not know where the player is looking so the depth of field cannot change depending on that. But there are ways of making it interactive according to Kass et al. (Kass, Lefohn and Owens 2007) and this could be used in other interesting ways.

For example an interactive depth of field which works the same way as the forced camera movement archetype described earlier, only that it sets the depth of field to this important object instead. This would however need to be investigated first. Does it confuse the player? Does it cause simulation sickness (Lan van der Linde 2004)?

Field of view
This is a static value in games which is either hardcoded or can be changed with the use of configuration files. In the games studied the value used is hidden in every game except for Tera Online which uses a value of 60. Many of these games are also released onto both PC and other platforms such as PlayStation 3. This can become problematic if the game is not created with different fields of view in mind as players can get motion sick according to Artsy Gamer (Artsy Gamer 2010).

With that in mind it makes no sense to not allow the players to configure the field of view themselves. If the players get motion sick playing the game it is unlikely that they will finish the game, buy downloadable content or buy the sequel.
**Lens flare**
Out of the games in this paper, Batman Arkham City, Tera Online, Tomb Raider and Sleeping Dogs (Figure 21) uses lens flares. Four games out of six.

![Figure 21: lens flares in Sleeping Dogs.](image)

Computer generated lens flares have existed for a long time and many designers either hate or love them, some use them a lot and others don’t use them at all. The director of the Star Trek movie released 2009, Abrams, used a lot of them while still thinking they were ridiculous. While some designers may hate the effect and decide to not use it, that does not make it any less realistic.

**Motion-blur**
A big difference in how motion-blur works in reality and in games is that it is applied in situations when the designer says it should be used than depending on what speed objects are moving. To make an example there is no motion blur when the avatar in Assassin’s Creed 3 jumps from a one hundred meters high tower, but there is if the avatar stabs someone.

Every game in this paper uses motion blur but the effect is applied in different ways. In Batman Arkham City it is applied when the player moves the mouse around quickly (Figure 22), which often happens in combat for example. On the other hand in Sleeping Dogs, the motion blur is applied when the avatar performs more powerful attacks.
Figure 22: motion blur applied during rapid mouse movement in Batman Arkham City. This could possibly be categorized as a whip pan movement.

Depending on the designer’s vision there are many ways of implementing motion blur, however none of these games has implemented it realistically by calculating the speed of which an object moves, in which direction and how long the virtual camera shutter is open. Rather it is used as a post process effect applied depending on the game state in these games.

**Radial Blur**

This effect is found in the configuration files of both Tera Online and Batman Arkham City, the games made using Unreal Engine 3. However no form of radial blur has been observed within the gameplay of Tera Online. Assassin’s Creed 3 on the other hand continually uses a very strong radial blur effect in combat for finishing moves as in figure 23. It is also often paired up in combination with a slow motion effect, possibly to emphasis the action. It is used in a similar way in Batman Arkham City.
Implementing this feature realistically might be hard as it requires the camera to move at a fast speed. Using it as effect when the avatar stabs an enemy should be considered a very stylistic use as it’s not even close to reality unless the camera would be strapped to the avatars hand or the like.

However both Batman Arkham City and Assassin’s Creed 3 have game moves involving the avatar falling or jumping down from heights at very high speeds. What is interesting about this is that neither actually uses the radial blur effect for this occasion (although Batman Arkham City utilizes a similar looking post processing effect for this). It might have been the designer's decision but the implementation and decision of when to use radial blur is not very realistic.

Tera Online on the other hand does have the capability to use radial blur as the game also uses the Unreal Engine 3 but the designer has opted not to. As the game already arguably uses the heaviest post processing effects, bloom and depth of field in particular, compared to the other games it could intentionally have been left out in order to optimize performance.

**Ambient Occlusion**

The games from this study using screen space ambient occlusion are Assassin’s Creed 3, Tomb Raider and Sleeping Dogs. Batman Arkham City and Tera Online are also using some form of real time ambient occlusion algorithm but which one is unspecified. However Epic Game’s documentation of post processing ambient occlusion versions (Epic Games 2012) implies that they are also using screen space ambient occlusion as the feature was added the same year that Crytek created it. If this is the case, five games in this study are using screen space ambient occlusion.
Upgrading the real time ambient occlusion algorithm is desirable for any engine still using the 2007 screen space ambient occlusion algorithm as some of these new techniques can be less performance heavy as well as producing better results. Other techniques simply innovate on the already existing, Mattausch et al. (Mattausch, Scherzer and Wimmer 2009). While it is not public information when Crystal Engine (Tomb Raider) had its algorithm updated last time, Unreal Engine has not seen an update since 2008.

5.4 Camera Movements Discussion

Aerial establishing shot
The establishing shot is used for establishing the context between different subjects in the scene and to introduce the viewer to the scene where events are about to happen according to Prunes et al. (Prunes, Raine and Litch 2002), although they might not always be aerial. However the aerial establishing shots have been more common in the games studied whereas establishing shots taken from the ground have been more of a result of the player entering an area from a specific direction.

The aerial shots however have often been intended since they are mostly connected to game moves. There are a few exceptions such as the avatar starting a level on top of a mountain and then descending to ground level (Tomb Raider). It is also debatable if this truly counts as an aerial shot as the camera is still close to the ground albeit at a high altitude and looking down on the surrounding area.

Establishing shots from the ground were only really used often in Sleeping Dogs, but this is mainly because the game almost completely lacks any form of vertical movement. The player might climb some stairs and end up on a rooftop once in a while but it is nothing compared to the aerial shots of the other games.

Either way, designers could make more use of both aerial and grounded establishing shots in many games to give the player an idea about what the area looks like. They were barely used at all in Dark Souls but this might be a design decision to keep the game scary by keeping the player from knowing what is up ahead. But if the game does use establishing shots it could be beneficial for the designer to use different kinds instead of focusing on one, such as the aerial establishing shot.

Arcing default camera around the avatar at a fixed distance
Unless the camera model is changed completely to one with static positions or some other model, it will be difficult to change this a lot. Some games change the distance in different game modes but others keep it completely static. But even if the distance is changed in different game modes the player might still remain in one game mode for longer periods of time which still makes the camera more static.

This could be remedied by having a script which is always active which changes the distance and maybe rotations as well. As long as the script does not interfere with the player’s actions or other important camera movement it would make the camera movement more dynamic. Still, some scenes might require a completely static camera but a few exceptions would not be a problem.
As for the rotation part of this archetype, the games have solved it very differently. Some games allow free rotation and tie movement to the camera rotation while other games try to always focus the camera behind the avatar. This design choice should relate to gameplay rather than aesthetics since the player might need more or less information about their environment. For example while fighting an NPC in Tera Online the players can still look around themselves to look for other threats if necessary.

**Camera distance tied to avatar position**
This is one of the ways that the games make the cameras more dynamic. While not every game uses it, others can make much use of it. Batman Arkham City for example allows the player to go from high grounds to low grounds and back in a matter of seconds which makes it more noticeable.

The camera distance change can also be triggered by different types of position changes. In Batman Arkham City it can zoom out if the avatar is standing on a high ground or zoom in if the avatar is crouching. In Sleeping Dogs on the other hand it is instead triggered depending on the avatar’s position in the world on the horizontal plane, in which area the avatar is currently at.

Other games could probably also use this type of camera movement since they also have a wide variety of areas with different themes where it might be important to emphasis the height (climbing mountains in Tomb Raider) or how confined the space is (descending the catacombs in Dark Souls).

This is also slightly related to the part where cameras normally move closer to the avatar should any geometry be in its way. Some developers might have been lazy and simply used this feature to create a closer shot in confined areas. So the camera relies on the geometry behind it to stay close to the avatar instead of a script telling it to because of the area itself.

**Camera distance tied to avatar velocity**
The other common way of changing the distance of the camera is depending on the avatar’s velocity. This can include for example increasing the camera’s distance to the avatar when the avatar is running, jumping or traveling in a fast vehicle. Sometimes this is also used in combination with a camera shake effect. Because there are many different ways to travel in these games there are also many different ways to invoke this type of camera movement.

This gives the benefit of creating a more dynamic camera when gameplay is very one-sided. Having the camera move back and forth a bit depending on how the player is moving is any easy way of achieving some form of camera interactivity depending on the situation. It is only used in three games out of six though and seeing how dynamic Tomb Raider’s camera is normally there does not seem to be a reason for the designer to not have implemented it.

**Forced camera movement**
While not used as often as the other archetypes it is instead used in many different games, often in conjunction with what would else be a short cut scene. This kind of shot is also capable of taking the camera out of its normal arcing behavior which creates a contrast if the designer made the shot different enough. The drawback is that the avatar’s movement might not be what the player expects if the direction of movement is tied to the direction of the camera.
**Locked over-the-shoulder shot targeting systems**

The targeting systems are mainly made for gameplay and their primary function is to allow the player to more easily attack the target or dodge the target’s attacks. As the avatar’s movement is locked in relation to the enemy and the avatar is always facing that enemy it becomes very hard to actually miss with the attacks.

The targeting systems themselves in Dark Souls and Sleeping Dogs works almost exactly like this; however the camera work in Sleeping Dogs is not always an over-the-shoulder shot. Assassin’s Creed 3 on the other hand uses an automated targeting system which will lock an enemy but it is not a player action, and the camera might also move around sometimes.

Unless it is very critical that the player understands what enemy they are targeting instantly, there should not be a problem with using something which is not an over-the-shoulder shot. Dark Souls is the only game strictly forcing this camera view and this might be a choice by the designer to clarify what enemy is the target (always centered on the screen).

There might not really be a reason for other games to implement this form of targeting system either as they have mostly solved the problem in their own way. In Barman Arkham City the avatar will attack in a general direction and then more or less fly towards the enemy in that direction to land the hit, while Tomb Raider focuses more on ranged attacks and aiming whereas the few melee attacks are quite wide and hard to miss with.

**Over-the-shoulder shot while aiming**

All games except Dark Souls and Tera online uses this type of aiming in a very similar fashion. Dark Souls do use the same form of aiming with close to the same positioning of the camera and a crosshair but the camera is too close to the bow or crossbow to be called an over-the-shoulder shot.

A problem with third person aiming is how it actually works gameplay wise. What appears to be a perfectly fine target from the camera’s point of view might make the avatar shoot straight into a wall. So changing the camera’s position to an over-the-shoulder shot, and even closer to the weapon in Dark Souls, does have a purpose.

The only game which does not change the position of the camera is Tera Online and this makes aiming a bit harder than in other games. To be clear the camera in Tera Online rarely moves at all unless the player commands it to but this is one of the instances where it could help the player.

Since the player’s aim often is very vital to the gameplay the designer should not be too creative with the aiming mechanism. An over-the-shoulder shot every time might not be the most interesting way of doing it but it ensures consistency, letting the player aim in peace.

**Partial camera control**

Many of the games use this technique to add interactivity to something which would otherwise only be a cut scene. If the player is at least allowed to control the camera a little bit, she is still somewhat in control of the game. Since the technique is easily combined with other techniques it is also possible to integrate it in many other different situations. So for example it can be used in cut scenes, during scripted events or during specific game moves.
Even so there are many of said occasions where the technique is not used where it could have been. An example would be some shots during finishing moves in Sleeping Dogs which uses a camera which is completely scripted and does not allow the player to control it at all. It is hard to believe that a game move needs an exact angle to be aesthetically pleasing or to show that the enemy is in fact being finished off. Adding this effect would add a layer of interactivity where else there would be none unless you count the pause button.

**Player controlled post-processing effects**
This is always tied to some form of game move which is meant to clarify to the player what they can do, expect or interact with. Basically it is used instead of giving hints to the player to let them know which button to press or what enemy got a powerful weapon. As it is used in three out of six games it is hard to see it as something innovative, especially since it is a game move and not an event, game state or cut scene.

The game move in one game is very similar to those of the other games and thus not very unique. It might be hard to use another technique and achieve the same results in terms of gameplay but right now the game moves are resembling each other a bit too much. Players will surely notice the similarities so another solution would make the game stand out more.

**Relative positions and angles for game moves**
If the game move plays an animation which ignore the player’s commands until the animation has finished, this is one of the few occasions where the camera can actually cut to a new position. And since the event is going to be played out the same no matter what input the player gives, this is one of the few times it is possible to cut the camera without interrupting anything that the player might be doing.

Additionally this movement can also expand on the usual arcing shot to make the camera movement more interesting as the designer has more freedom. Something which is used a lot in these games. The drawback once again is that this type of movement seldom is interactive. This technique should be combined with the partial camera control as often as possible to allow the player to maintain some form of control over the game.

### 5.5 General Discussion

There are many points on which the games presented in this study can improve on their virtual cameras, either technically or aesthetically. While some game engines simply lack a specific feature other do not use it when the game engine has it. Other games use these effects in a stylistic way but still try to emulate reality in other aspects, which can be quite contradictory depending on their design goals.

The usage of these features could also be a lot more creative and mimicking the film industry would be beneficial in several ways. Techniques which could have been used but have not includes whip pans, any shot which changes the depth of field such as dolly zoom, or rolling the camera. Some other features could also be considered standard to implement in games, motion blur, bloom and camera shake effects were used in every game.

The actual camera movement is often worse when it comes to innovation. Some games like Tera Online and Dark Souls have close to static cameras which are terribly uninteresting, while other games such as Tomb Raider has a camera which rarely stays in one place. Even so there is improvement which could be made to the camera in Tomb Raider as well.
Some of the games use very similar default cameras when the avatar is moving around. In general the camera arcs around the avatar but using different kinds of scripts and changing the camera angle and position depending on the game state it is possible to create a more interactive camera. Most of the games also have very similar camera techniques for similar gameplay which does not help them stand out. These include the player controlled post processing effects and over-the-shoulder shots while the avatar is aiming.

Improvements which can be made for most of these games include but are not limited to: letting the player always retain some form of control, making the camera more dynamic by interactively changing the camera’s distance to the avatar or using more different angles for shots such as the establishing shots instead of using the same one over and over again. Then there are also other forms of filming techniques which have not been used such as forced perspective.

Unless the designer responsible for the camera has experience in the film industry it might be worth hiring a cinematographer. Some of these games have used techniques such as rolling the camera but only during cut scenes or cinematic sequences. This implies that the person in charge of the camerawork for those scenes has more experience than the designer for the camera used during gameplay as the camera in the cut scenes use more techniques. Having a cinematographer look into the camerawork used during gameplay would surely help elevating the quality of the game. Of course there might also be a conflict with the design of the game where a certain type of camera shot simply cannot be implemented.

6 Conclusion

In this study six games have been analyzed in terms of camera technology used and camera movement used. Data was collected by gathering information from the game’s graphic settings menus and configuration files as well as by observing them either by watching videos of the game, watching someone else play or playing. The camera movements were then categorized into different archetypes and the data collected from the games was used to place them into tables showing what technology and movement had been used.

This was then discussed and some improvements for these games and games in general have been suggested to make the camera work more interesting. The focus is mainly interactivity and the usage of other techniques from the film industry. The most important points are: increasing interactivity by letting the player keep control to some degree of either the camera or avatar at all times, breaking away from these archetypes to make the game more unique, copying techniques from the film industry which are new to the game industry, and upgrading the old screen space ambient occlusion algorithm.

The subject has only been lightly touched upon and there more further research which could be made, in particular there is far more camera movement in games than what has been categorized into these archetypes. Only some of the most common camera movements were studied in this paper. The number of games could also be increased for a more quantitative analysis. Something else which could be of use is research about adapting filming techniques for the interactivity of games.
References


Rocksteady Studios, 2011. Game Info: Game Details. [online] Available at: <http://community.batmanarkhamcity.com/game/gamedetails> [accessed 17 May 2013]


**Appendix**

**Vocabulary**

Aerial Shot – A shot taken from far. In reality this would use for example a helicopter or a crane.

Arcing Shot – The camera moves around the subject while rotating to keep it centered.

Avatar – The character in the game that the player controls.

Close up – A certain part of the subject is shown in the frame, such as the face of a human.

Dolly zoom – Moving the camera closer while zooming out the lens or vice versa.

Establishing Shot – This shot shows the viewer where scenes will soon unfold.

 Forced perspective – Optical illusions used to make objects appear closer, farther away, smaller or bigger.

 Game State – This is the current state of the game. Depending on it the player can take different actions and the camera will behave accordingly to the script used in this state.

 High Angle Shot – A shot taken from above.

 Lens Effect – An effect in filming created using the lens.

 Low Angle Shot – A shot taken from below.

 Medium shot – Most of the subject is inside the frame.

 Over the Shoulder Shot – A shot from slightly behind and over the shoulder of a person.

 Point of View Shot – The camera is placed where the eyes of a person should be.

 Post Processing Effect – effects applied after rendering but before output.

 Two Shot – A shot framing two people, there are also Three Shot and so on.

 Virtual Camera System – One or more scripts which decides how to use the camera.

 Whip pan – quickly changing the direction of the camera, causing motion blur.

 Wide shot – The subject is fully inside the frame.