Entertainment Computing 27 (2018) 188–193

Contents lists available at ScienceDirect

Entertainment Computing

journal homepage: www.elsevier.com/locate/entcom

A novel parallax engine for animation using hybrid graphics software

Sunil Kr. Jha[a,b,*], Stefan Shorko[c]

[a] Department for Management of Science and Technology Development, Ton Duc Thang University, Ho Chi Minh City, Viet Nam
[b] Faculty of Applied Sciences, Ton Duc Thang University, Ho Chi Minh City, Viet Nam
[c] University of Information Science and Technology "St. Paul the Apostle", Ohrid 6000, Macedonia

ARTICLE INFO

Keywords:
Graphics
Animation
Parallax
Simulation

ABSTRACT

Three-dimensional graphics software has been extensively used to create animations, video games in multimedia applications. However, the graphics software requires user's efficiency and long creation time in order to maintain a perfect animation. In the present report, a novel parallax engine has been proposed with the objective to ease the designer's work into instantaneous and effective outcomes. The proposed parallax engine achieves accuracy up to 99.97%. An average error rate of 0.03% was obtained in the motion tracking of objects in different layers of animation. The location of objects in layers of animation has been measured in the arbitrary unit.

1. Introduction

Visualization is an effective learning technique to recognize the objects by involving the viewer thoroughly using the eye to brain's perception [1–4]. It is helpful for explicit resolutions, such as describing the difficult topic, introducing specific maneuvers in a creative style, presenting the proper guidance to follow technical procedure, and circulation of information, etc. [1]. Moreover, visualization can also be used for improving memory, restoring health, reducing stress, and increasing motivation, etc. [1–3]. Computer graphics assist in visualization of facts by using images. Commonly, computer-generated imagery has been used in the creation of games, movies, modeling, and designing of catalogs, etc. [5–7]. Consequently, the knowledge of computer graphics generation is necessary for modeling of an animation engine.

Several experiments have been conducted in the past for the development of game engines and testing their correctness in terms of precision and smoothness [8–16]. Some of them are as follows: Bishop et al. [8] have designed a PC game engine (NetImmerse) with and without 3D graphics acceleration with a high-level programming interface; A novel game engine for ray tracing in the urban environment has been suggested using raster digital elevation method [9], which contains valuable features, including 3D modeling, accurate measurement of ray parameters of the different station in the urban environment, and low computation time, etc.; Moreno-Ger et al. [10] have developed a general game design method for online education, comprising adaptation and assessment features; A game engine for urban search and rescue domes has been designed by Wang et al. [11] to construct precise simulation for single robot and group.

The role of game engines in an animated movie is very simple but effective as it can handle high-end graphics and animations in less rendering and production time, which is the main issue, at the present time. Besides that, it can also be used for adding visual effects. A detailed report on how to use the game engine in developing animation can be seen in Ref. [12]. The data-driven speech animation has a significant contribution to improve the quality of animation. For example, a statistical model [13] in which phoneme-based speech animation trajectory fitting metric was generated, and then a regression model was used to learn the association. In some other related studies, a real-time animator for hybrid systems using the finite automata with differential equations has been developed [14]. The mesh denoising method has been implemented to recover high-quality graphics [15], and a multi-level graphics processing system was proposed for animation [16], etc.

Computer graphics are derived into raster which uses pixels to shape an image using the available raster programs [1] and a vector which utilize lines, shapes, and text to create a more complex image. However, some software can only generate vector graphics, like Inkscape, and the other can generate only pixel graphics, like GIMP [17]. Consequently, there is a necessity of hybrid software which can work with both types. This is the main motivation behind the present study to develop a parallax engine using the hybrid software. The Adobe Flash is a standard software used in the proposed engine as it contains the scripted codes for automatic tasks. It is substantial to identify the

* Corresponding author at: Department for Management of Science and Technology Development, Ton Duc Thang University, Ho Chi Minh City, Viet Nam.
E-mail address: drsuniljha@tdt.edu.vn (S.K. Jha).

https://doi.org/10.1016/j.entcom.2018.07.001
Received 29 December 2016; Received in revised form 10 July 2018; Accepted 11 July 2018
Available online 12 July 2018

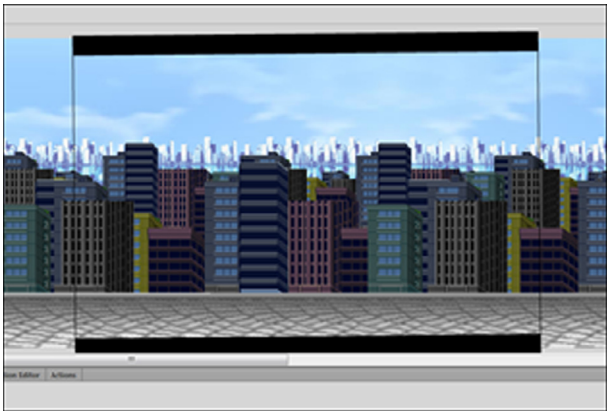

Fig. 1. The proposed parallax engine in Flash 8.

elementary features of Adobe Flash, like handling movie clips, and posting multiple symbols within a symbol, etc., and understanding of scripting languages (Action script version-2 (AS2) or AS3) is preferably necessary for better use of the engine in animation [18]. The proposed parallax engine in the present study considers the key-frame motion. The distance between objects and motion trajectories were used in evaluating the speed of objects at each frame. Some other issues relevant to parallax engine which need to be addressed in the future include, a proper modeling of an animation engine so that it will result in an automatized parallax view in the 2D space [14,19,20].

2. Parallax engine

2.1. Origin of proposed engine

The Flash software was developed by Macromedia (the first version being Flash 8). Although the proposed engine is developed for Flash 8, it is also compatible with newer versions of Flash. Fig. 1 demonstrates a simple look of the parallax engine in Flash 8.

A new instance of a rectangular object was created to overshadow the canvas as shown in Fig. 1. It manipulates space freely without considering the location of the object on the canvas. Meanwhile, it is tough to view the canvas behind objects. One feasible solution is to use a transparency filter to all the objects, which is a tedious task, especially if the work is done in the frame by frame (FBF) format [8,18,21]. By comparing the Flash's standard GUI and the script-modified GUI with the virtual camera (Fig. 1), it is evident that one can view the borders of the preview and move it around the objects without manipulation which reduces the production time and increases the quality of animation.

2.2. Comparison amongst the flash, virtual camera, and parallax scroller

The differences by using the software manually, the virtual camera and parallax scroller individually to create an animation are described in following sub-sections.

(a) *Using the software manually*: There are multiple tools available in Flash, such as the motion and shape tween, the object notation, the real-time graphics tracker, the timeline access, and the sound libraries, etc. to create an animation. Fig. 2(a)–(d) represent the interface of the software, specifically, Fig. 2(a) shows the canvas. Sometimes, objects in the animation overshadow the borders of the canvas (Fig. 2(b)). There are two possible solutions to overcome the earlier constraint. First by using the transparency filter to make the object less visible [22,23]. However, adding the filter is not an easy task, since the precedence of positioning the objects is also adding up the complexity. For instance, if there are objects in layers above the background, they will all blend, and again overshadow the canvas to an extent due to multiple objects (Fig. 2(c)). Second by using the layer outline method, which will allow only the borders of the object to be visible [24,25]. Again, this method is not fully appropriate because one cannot orient by using borders only (Fig. 2(d)). The detail steps to create an animation in Flash by using either AS2 or AS3 can be seen in Ref. [18].

(b) *Using the virtual camera*: Adjusting the canvas behind the objects is a tough task in the manual method. The application of virtual camera solves this issue. Meanwhile, the coordinates of the canvas can be measured through the motion of an object by using the virtual camera. The main difference between the two scenarios is that the virtual camera reduces the production time compared with the manual production. The virtual camera projects the objects regardless of the position of the canvas and the direction of the camera. It is always on the topmost layer for easier orientation. Although this method is more efficient, it doesn't accept distortion such as skewing or rotation.

(c) *Using the parallax scroller*: The application of virtual camera requires the information of scripting languages for the animator. Also, the parallax scroller offers many other advantageous features. For example, it transfers the objects on its own while the virtual camera doesn't support automatic parallax. In addition, the way parallax scrolling is maneuvered is complicated in order to create animations. The animation and programming in scripting languages need to be combined in such a way that will eventually get the optimal resolution, which is actually a challenging task. It is further explained in the next section.

2.3. An overview of proposed parallax engine

The main differences between the proposed engine and the prior techniques, like the virtual camera [26,27] and the parallax scroller [28,29] are as follows. The prior techniques friendly only for developers, i.e. one needs to work on codes and handle the parameter values, variables, and compound functions, etc. which are tough to identify. The earlier two techniques are contrasting as well as complementary. Meanwhile, the proposed engine is appropriate for animations, and other specific works, like video games. The proposed engine applies the certified virtual camera and the parallax scroller altogether in such a way that the user doesn't have to concern about the programming code at the behind. The engine functions, like a special type of virtual camera (parallax camera or parallax engine), which can be used for 2D and 2.5D graphics. The code of the proposed parallax engine is composed in a linear way, as it parses the coordinates of the objects automatically, and the user has to move only the parallax camera. Manually, the user needs to do every single thing on the timeline, while in the parallax engine there is a predefined template in which the user needs to change the objects only. The parallax engine can perform every single task behind a virtual camera which withholds a synchronized code for the parallax scroller. Another significant feature of the proposed engine is that it can be simply inserted as a plugin, whereas the virtual camera and parallax scroller is needed to be configured by checking the codes. The less production time is the most important feature of the proposed engine. The application details of the proposed parallax engine will be described in detail in the next section.

3. Experiments using the proposed parallax engine

The experiments were based on two main variants (i) frames per second (FPS), and (ii) number of layers essential for the engine. The variation in the value of FPS was from 24 to 30 (For instance, 30 s of animation with 30 FPS is equal to 900 frames) and the number of layers was changed from 1 to 8. The objective of these experiments is to compute the production time manually and using the proposed parallax engine for comparing their performance. Manually, for 30 FPS and 7