Table of Contents

International Journal of Computational Methods in Heritage Science

Volume 1 • Issue 1 • January-June-2017 • ISSN: 2473-5345 • eISSN: 2473-5337
An official publication of the Information Resources Management Association

Research Articles

The *International Journal of Computational Methods in Heritage Science* is indexed or listed in the following: Cabell's Directories; Google Scholar; Ulrich's Periodicals Directory

International Journal of Computational Methods in Heritage Science
Volume 1 • Issue 1 • January-June 2017

Realistic Simulation of Cultural Heritage

Chairi Kiourt, School of Science & Technology, Hellenic Open University, Patra, Greece
George Pavlidis, Athena Research Centre, University Campus at Kimmeria, Xanthi, Greece
Anestis Koutsoudis, Athena Research Centre, University Campus at Kimmeria, Xanthi, Greece
Dimitris Kalles, School of Science & Technology, Hellenic Open University, Patra, Greece

ABSTRACT

One of the most challenging problems in the simulation of real environments is to generate worlds that appear realistic and more attractive. It becomes increasingly challenging when the simulated environment focuses on minors (students), because the young generation has high demands on simulation systems due to their experience in computer gaming. Virtual museums are among the most important simulation environments, which present cultural and educational content for everyone. Their purpose is to enrich the users experience by allowing an intuitive interaction with the museum artifacts and to offer knowledge with the most pleasant ways. This paper focuses on the aspects of realistic simulations in the development of virtual 3D environments for Cultural Heritage applications. This study includes aspects regarding some of the most high-tech image effects, applicable artificial intelligence methods, powerful game engines, how real object can be reconstructed realistically and how all those features may be combined to produce realistic, pleasant, productive and educative environments.

KEYWORDS

Cultural Heritage, Gamification Realistic Environments, Interactive Artificial Intelligence, Realistic Simulations, Virtual Museums, Visual Realism

INTRODUCTION

In realistic simulations of Cultural Heritage, virtual museums have a prominent role. Virtual museums are virtual environments that host virtual exhibitions created to tell the stories of the real museum artifacts to their visitors, just like in real museums, in a richer context and a wider range of possibilities. In the context of a real museum, this is achieved by combining exhibits and information in a carefully designed order and presentation style (Lepouras & Vassilakis, 2004). In the creation of virtual museums, contemporary IT technologies, such as multimedia, 3D computer graphics, spatial sound and virtual reality, can be used to enhance the presentation (the virtual presence), offering a more vivid and enjoyable experience. Although each of the different technological innovations involved pushes towards their own respective, there are two main goals that are universal in such systems: *realism* and *speed*. In addition, the continuous development of Web services and computer infrastructures complemented by the increasing availability of computer game development platforms (also known as *game engines*), contribute towards a continuous release of serious games in diverse fields including entertainment, cultural heritage, education, artificial intelligence, sociology, military and health systems (Breuer & Bente, 2010). In a sense, serious games can be considered as an efficient approach for blending domain specific activities, like in cultural heritage and education, with gaming. By utilizing contemporary visualization and simulation technologies, serious games enhance the

DOI: 10.4018/IJCMHS.2017010102

user's experience through photorealistic interactive environments (Van Eck, 2006). This form of stimulation is considered to be one of the primary factors for successful user engagement, in which playing, assumes the role of the driving force that promotes concentration in the activity process and user encouragement for further self-improvement. Stimulation is largely considered by the scientific community as a desired mechanism to achieve the desired results (Rogers, 1996).

Numerous works utilize various technologies to provide solutions for history teaching and learning, or to enhance actual museum visits. The notion of virtual museums and exhibitions has been introduced as an approach to overcome the limitations of the physical space and to provide a vivid experience to remote visitors (Tsichritzis & Gibbs, 1991). An overview of virtual museum technologies is presented in (Sylaiou, Liarokapis, Kotsakis, & Patias, 2009). The Augmented Representation of Cultural Objects (ARCO) (Patel, White, Walczak, & Sayd, 2003) (Wojciechowski, Walczak, White, & Cellary, 2004) was among the predominant efforts towards a dynamic virtual museum system accompanied by a 3D digitization technique to provide a framework to produce and exhibit 3D digital replicas of their artifacts. ARCO was also, evaluated as an Augmented Reality Interface (ARIF) (Sylaiou, Mania, Karoulis, & White, 2010) by focusing on the user experience in relation to the technologies of ARCO. In 2004, Lepouras & Vassilakis (2004) presented the concept of creating virtual museums focused primarily on educational content and related services by using a game engine. This virtual exhibition space took advantage of the high visual quality of modern game engines. Pavlidis et al. (Pavlidis, Tsiafakis, Provopoulos et al., 2006), proposed a Web-based 3D digital replicas management system with a dynamic virtual exhibition showroom. In addition, in (Pavlidis, Makarona, Arnaoutoglou et al., 2008) a more advanced framework for digital museums has been presented, where a non-photorealistic digital replica of a real museum is used to demonstrate educational activities rather its actual exhibition, aiming at increasing the museum's visitors. Sookhanaphibarn & Thawonmas (2009) presented a 3D virtual museum developed in the Second Life 3D world engine. The virtual museum was equipped with an innovative intelligent guidance system that was able to provide a customized navigation route based on the visitors' preferences (2009). In (Djaouti, Alvarez, Rampnoux, Charvillat, & Jessel, 2009) an interactive serious game is used for the promotion of a prehistoric heritage site. Anderson et al. (Anderson, McLoughlin, Liarokapis, Petridis, & Freitas, 2010) reviewed the state-of-the-art of theories, methods and technologies utilized by serious games as cultural heritage promotion tools by showing case studies that exploit such technologies. Papastamatiou et al. (Papastamatiou, Alexandridis, Tsergoulas et al., 2010), presented a dynamic Web-based 3D e-shop system for commercial use, offering a WYSIWYG graphical user interface. Furthermore, Sillaurren & Aguirrezabal (Sillaurren & Aguirrezabal, 2012), presented 3DPublish, a content management system that was developed in Unity3D game engine and allowed the generation of dynamic 3D exhibitions. Furthermore, works like (Bellotti, Berta, Gloria, D'ursi, & Fiore, 2012) focus on a generalization of the task-based learning theory in applications using smart mobile devices. In addition, Koutsoudis & Pavlidis (2011), proposed a novel approach for navigating within complex cultural scenes by exploiting content-based retrieval descriptors. Koutsoudis et al. (2012), proposed a content-based navigation framework for a virtual museum, based on metadata that describe the exhibits and thus providing a semantic similarity-based navigation. In (Doulamis, Liarokapis, Petridis, & Mialous, 2012) a serious game is proposed based on scenarios derived from the cultural heritage domain and attempts to enrich the player's knowledge by spreading a mystery in ancient times. In a recent work (Bonis, Vosinakis, Andreou, & Panayiotopoulos, 2013) proposed a multi-user framework for virtual exhibitions that adapts to visitors' preferences. Recently, Kiourt et al. (2015a; 2015b; 2016) have studied many aspects of interactive virtual museums mainly focusing on interactive 3D object exhibitions with dynamic open data content; the virtual museum was re-introduced as a cultural and educational tool that focuses on user-driven exhibitions.

In all most all the aforementioned approaches, as in many other – a small list of virtual museums for educational purposes can be found at Sylaiou et al. (Sylaiou, Liarokapis, Kotsakis, & Patias, 2009; Sylaiou, Mania, Paliokas et al., 2016) – the main focus is to offer users pleasant and realistic virtual environments that try to simulate the real museums. The main contribution of this study is to introduce and present some of the most important factors that should be taken into account in building realistic simulations of Cultural Heritage that are accepted by all ages (children, adults and seniors). The study takes into account previous studies such as (Yu, Mortensen, Khanna, Spanlang, & Slater, 2012; Callieria, Debevecb, Pairc, & Scopig, 2006; Prem, Thalmann, Moccozet et al., 1998; Papagiannakis, L'Hoste, Foni, & Thalmann, 2001; DeLeon & Berry Jr, 2000; Addison, 2000). The study is presented as follows: (a) the most effective computer graphics techniques for building visual realism are being presented, (b) some of the most important artificial intelligent technologies applied in virtual museums are being reviewed, (c) game engines are being introduced and analyzed, (d) realistic reconstruction of the virtual environments is being presented, (e) the impact of the virtual environment/museum in education is being commented. The study concludes with some of the key remarks and highlights.

EFFECTS FOR ENHANCED REALISM

In order for the 3D virtual environment to be more realistic, some real-time image effects are being used, which significantly affect the quality and efficiency of the experience and do not impose any additional hardware requirements (Yang, Dong, Alelaiwi, & El Saddik, 2015). Often those image effects are named as *Screen Spaceeffects*. It should be emphasized that by activating real-time image effects that are being recalculated for every pixel of every frame per second, the system exhibits high computational costs, which are only diminished by intelligent algorithms and the usage of sophisticated graphics hardware. A screen-space approach is one that involves 2D images of objects rather than the 3D geometry (Cohen & Wallace, 1993; Jonghyun, Jae-Hyun, Changwon & Byoungho, 2013). Generally, a 3D object or scene is represented in terms of geometric primitives such as vertices, polygons, edges, and surface patches. Graphics algorithms directly operate on geometric-space (also referred to as object-space) data to produce a desired result. On the other hand, image-space algorithms first rasterize the 3D geometry into one or more 2D images, and then operate on the pixels of these images. The rasterization step discretizes the 3D scene into pixels as visible from a given viewpoint. Each pixel in the image thus represents a finite region of the scene. Therefore, all the information pertaining to that region required to perform the rendering can be stored at the corresponding pixel location in the image, such as 3D positions, surface normal vectors, etc. In the following paragraphs the main screen space effects that contribute to the realism of a simulation are being presented.

Lighting, Reflection and Ambient Occlusion

Lighting is a method for calculating the shading of a 3D object, by using the intensity, direction and color of light that falls on it (Cohen & Wallace, 1993; Verbeck & Greenberg, 1984; Iones, Krupkin, Sbert, & Zhukov, 2003). Generally, nowadays, in almost all 3D environment developments tools, such as game engines, there are two types of lighting techniques: 1) *real-time lighting*, in which lighting is updated in every frame being displayed, 2) *baked lighting*, in which the lighting is pre-calculated and transform to a static lighting effect by being stored on the texture map of the objects, producing what is called a *light map*.

In order to cast light in a virtual 3D scene there are five types of virtual light sources that simulate real-world light sources; these sources are directional, point, area, spot and ambient light sources. A *directional light source*, as the name implies, illuminates an entire 3D scene from one specific direction, Figure 1a. It is generally used to represent lighting from a source at large distances from the scene. For example, sunlight is usually represented as directional light. A *point light source* (Cohen & Wallace, 1993) is represented with a single 3D point in the scene, generally placed in close proximity to the other objects of the scene, unlike the directional light source (Figure 1b). The rendering equation also simplifies under point light illumination since only a single illumination direction is used at each point, x. However, the direction changes with the position of the point. Perhaps the type of light source that mostly resembles the ones found in real life is the *area light source* (Cohen & Wallace, 1993). An area light source is defined as a fully-fledged 3D object, just like any other in the scene. It emits light over its entire surface area, therefore illuminating a particular point in the scene from multiple directions, unlike the point and directional light sources, in which a point is lit only from one direction (Figure 1c). The rendering equation can be adjusted to accommodate area lights by changing the integral over solid angle to integral over the area of the light source. Like a point light, a *spot light source* (Cohen & Wallace, 1993) has a specified location and range over which the light falls off. However, the spot light is constrained to an angle, resulting in a cone-shaped region of illumination (Figure 1d). The center of the cone points in one direction of the light object. Light also diminishes at the edges of the spot light's cone. Widening the angle increases the width of the cone and with it increases the size of this fade, known as the 'penumbra'. Spot light sources are generally used for artificial light sources such as flashlights, car headlights and searchlights. With the direction controlled from a script or animation, a moving spot light will illuminate just a small area of the scene and create dramatic lighting effects. An *ambient light source* is present all around the scene and doesn't come from any specific source object. It can be an important contributor to the overall look and brightness of a scene. An ambient light can be useful in a number of cases, depending upon your chosen art style. An example would be bright, cartoon-style rendering where dark shadows may be undesirable or where lighting is perhaps hand-painted into textures. Ambient light can also be useful if one needs to increase the overall brightness of a scene without adjusting individual lights.

In order to render more realistic images in computer graphics a model of how may objects reflect light is required (Cook & Torrance, 1982; Phong, 1973). The reflectance model must describe both the color and the spatial distribution of the reflected light. The model is independent of the other aspects of image synthesis, such as the surface geometry representation. Most real surfaces are neither ideal specular reflectors nor ideal diffuse reflectors (Cook & Torrance, 1982; Phong, 1973).

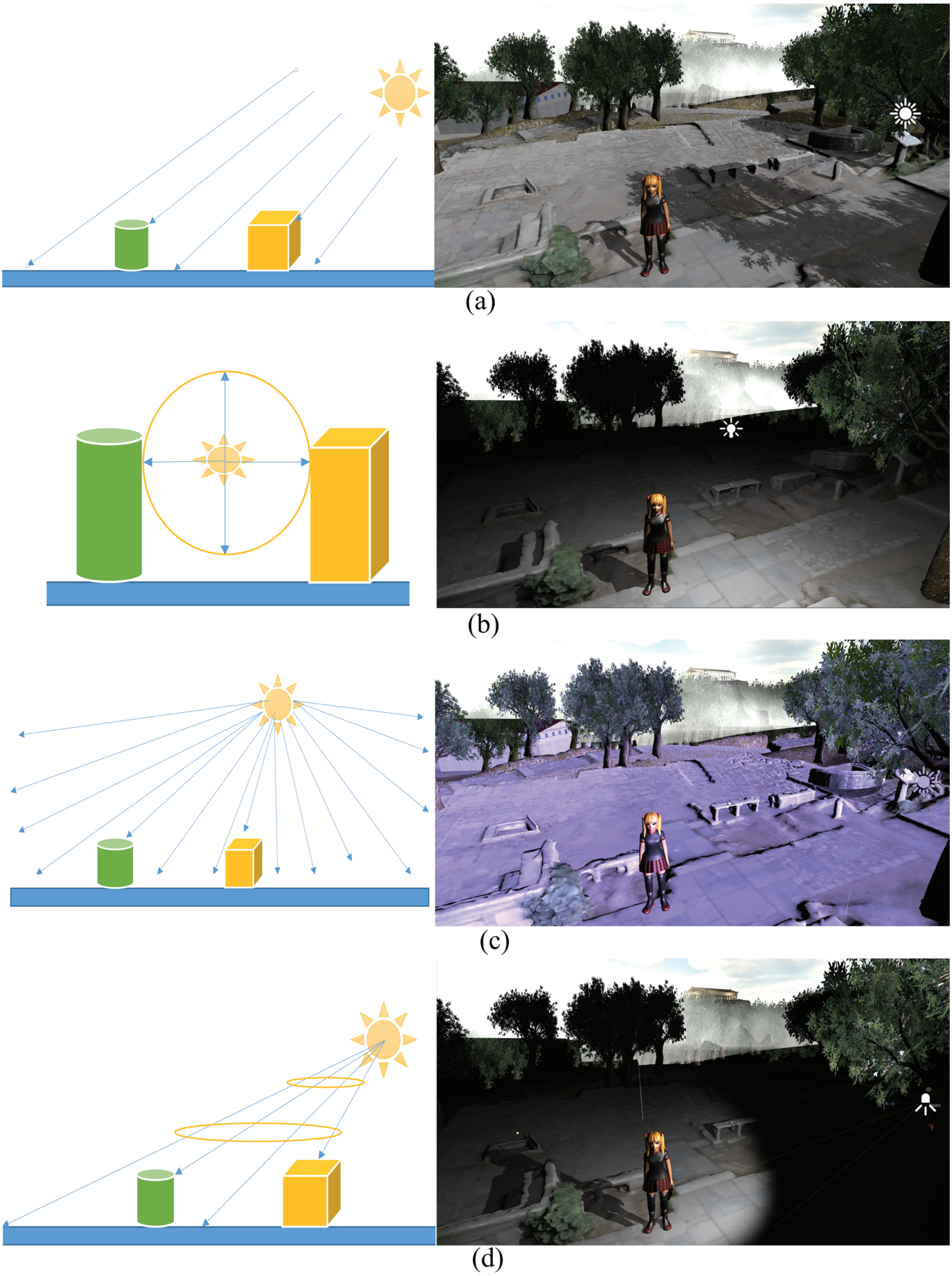
In any case, the general form of the equation that describes the basic lighting (Phong) model for objects in a 3D scene, in which multiple light sources might exist, is (boldface letters denote vectors)

$$I = K_e + K_a \times I_a + K_d \times I_d \times \max\left(\boldsymbol{n}\cdot\boldsymbol{v},0\right) + K_s \times I_s \times \delta\left(\boldsymbol{n}\cdot\boldsymbol{v}\right) \times \max\left(\boldsymbol{n}\cdot\boldsymbol{v},0\right)^s$$

where K_e the *emissive* part of light that comes from the object itself (might be zero); $K_a \times I_a$ the *ambient* part that corresponds to the effect of any ambient light source with I_a the global ambient light intensity (or color); $K_d \times I_d \times \max\left(\boldsymbol{n}\cdot\boldsymbol{v},0\right) = K_d \times I_d \times \cos\theta$ the *diffuse* part coming from a directed light, with K_d the object's diffuse reflection coefficient, I_d the light intensity and θ the angle between the light vector $\vec{v}$ and the surface normal $\vec{n}$ at the point of computations; $K_s \times I_s \times \delta\left(\boldsymbol{n}\cdot\boldsymbol{v}\right) \times \max\left(\boldsymbol{n}\cdot\boldsymbol{v},0\right)^s = K_s \times I_s \times \cos^s\phi$ the *specular* part coming from a directed light, K_s being the object's absorption coefficient, I_s the incoming light intensity, ϕ the angle that represents the deviation of view angle from mirror direction and the power s the shininess coefficient. The term $\delta\left(\boldsymbol{n}\cdot\boldsymbol{v}\right)$ results either 1 if $\boldsymbol{n}\cdot\boldsymbol{v}$ is greater than zero, or 0 otherwise.

Although lighting techniques provide support for all the basic calculations for the rendering of a visible surface element with respect to lights that might hit upon it and the surface characteristics, still, a realistic scene representation cannot be attained. This is due to an issue that relates with how exposed are those surface elements to the ambient lighting of the scene. That is, points that are occluded due to the geometry of their surroundings should receive less ambient light, thus should

Figure 1. Lighting types: (a) directional light, (b) point light, (c) area light, (d) spot light



appear darker in the final rendering. The solution to this issue has been encoded in computer graphics as *Ambient occlusion* (Yu, Wang, & Yu, 2010; Miller, 1994; Dutre, Bala, & Bekaert, 2006; Bavoil

& Sainz, 2009; Vardis, Papaioannou, & Gaitatzes, 2013). Ambient occlusion is a sophisticated ray tracing calculation, which simulates soft global illumination by faking darkness perceived in corners and at mesh intersections, creases, and cracks, where light is diffused (usually) by accumulated dirt and dust. Simply put, this technique darkens creases, holes and surfaces that are close to each other, as in real life such areas tend to block out or occlude ambient light, and hence they appear darker. A simple comparative example of using this technique in the presented system is shown in Figure 2.

There are several variations of real-time ambient occlusion, some of the most important are: Screen Space Ambient Occlusion (SSAO), Screen Space Directional Occlusion (SSDO), High Definition Ambient Occlusion (HDAO), Horizon Based Ambient Occlusion+ (HBAO+), Alchemy Ambient Occlusion (AAO), Angle Based Ambient Occlusion (ABAO) and Voxel Accelerated Ambient Occlusion (VXAO). The techniques that are based on screen-space computations target fast implementations by computing on the 2D projected visible scene pixels using pixel depth rather than the complete scene geometry. In general, the occlusion $AO_{\boldsymbol{p},\hat{n}}$ (or the surface irradiance $E_{\boldsymbol{p},\hat{n}}$) at a point $\boldsymbol{p}$ (in vector form) on a surface with a normal $\hat{n}$ at that point can be computed by integrating the visibility function $V_{\boldsymbol{p},\hat{\omega}}$ (or the incidence irradiance L) over the hemisphere Ω above and centered at the point with respect to the projected solid angle ω. The visibility function $V_{\boldsymbol{p},\hat{\omega}}$ equals 1 if the point $\boldsymbol{p}$ is occluded in the direction of the angle ω, or 0 if it is not occluded.

$$AO_{\boldsymbol{p},\hat{n}} = \frac{1}{\pi}\int_{\Omega} V_{\boldsymbol{p},\hat{\omega}} \cdot \left(\hat{n}\cdot\hat{\omega}\right) d\omega \quad E_{\boldsymbol{p},\hat{n}} = \int_{\Omega} L\cos\theta d\omega$$

Practically, SSAO for example, is implemented as a *pixel shader*, which analyses the scene depth buffer that is stored in a texture and for every pixel on the screen calculates the occlusion depending on the depth difference between the current and neighboring pixels.

Shadows play an important role in conveying reality in computer-synthesized images (Amanatides, 1987; Woo, Poulin, & Fournier, 1990; Hasenfratz, Lapierre, Holzschuch, & Sillion, 2003; Guennebaud, Barthe, & Paulin, 2006). A shadow may be described as a darkness region in an otherwise illuminated region (Woo, Poulin, & Fournier, 1990). Shadows also facilitate the comprehension of spatial relationships between objects. Though there are situations where shadows are not required (e.g. if the light source is behind the viewer or when modeling an overcast day), there are many other situations in which they are essential. The complexity of the shadows is related with the light mapping in trying to simulate the real environment shadows that are associated with the light sources (Amanatides, 1987). The presence of shadows makes 3D computer graphics look more natural. Without them, scenes often feel unnatural and flat, and the relative depths of objects in the scene can be very unclear.

Figure 2. A scene (a) with no ambient occlusion effect; (b) with ambient occlusion exaggerated

(a) (b)

They emphasize and clarify the three dimensional settings of the objects being displayed. However, the trouble with rendering high quality shadows is that they are computationally intensive and hard to generate. In computer graphics shadows appears in two types, first "hard shadows" are crisply defined with sharp edges and second "soft shadows" are less distinct and fade off toward the edges. An example of some hard shadows is depicted in Figure 3.

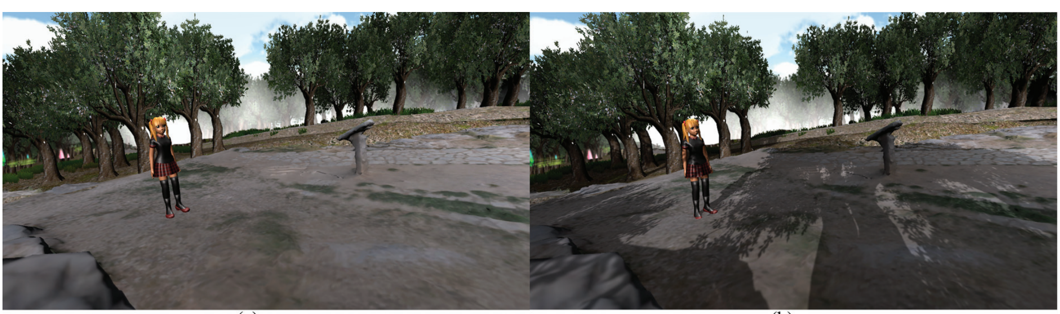
Depth of Field

Depth of field (DoF) is the distance between the nearest and farthest objects in a scene that appear acceptably sharp in an image or 3D scene (Yu, Wang, & Yu, 2010; Demers, 2004; Selgrad, Reintges, Penk, Wagner, & Stamminger, 2015). In computer graphics the simulation of DoF is a common post processing effect that simulates one of the most notable properties of a camera lens, the limited depth of focus. In real life, human eyes and a camera can only focus sharply on an object at a specific distance; objects nearer or farther from the camera will be somewhat out of focus. The blurring not only gives a visual cue about an objects distance but also introduces *Bokeh* (a Japanese word meaning *blur*), which is the term for pleasing visual artifacts that appear around bright areas of the image as they fall out of focus. DoF effects have been studied for several years. Starting around three decades ago (Potmesil & Indranil, 1981; 1982), there have been numerous approaches to adding DoF effects to computer graphics, but official published work in the area of DoF for computer graphics has been provided by Barsky et al (2003). The DoF simulation methods can be categorized in several ways in computer graphics (Barsky & Kosloff, 2008), but the two most important categories are: the *object-space* methods and the *image-space* methods. *Object space* methods operate on 3D scene representation, and build DoF effects directly into the rendering pipeline. *Image space* methods operate on images that were rendered with everything in perfect focus. Generally, *object space* methods generate more realistic results but image space methods are much faster (Barsky & Kosloff, 2008). Nowadays almost all game engines provide the *image space* methods. A sample of image space method applied in a 3D environment is presented in Figure 4. The left image depicts the application of the DoF effect for far away objects, whereas the right image depicts the application of the DoF effect for medium distance objects.

The mathematics of the blur due to a limited DoF in real-life photography are connected with the aperture of the lens and the *circle of confusion* (usually denoted as CoC), and are derived from trigonometry, usually by setting the far limit of the DoF to infinity, thus

$$CoC = \frac{f_c^2}{N \cdot \left(D_h - f_c\right)} \approx \frac{f_c^2}{N \cdot D_h}, \quad N = \frac{f_c}{d} \text{ the aperture, } \quad D_h \text{ : the hyperfocal distance}$$

Figure 3. A scene (a) with no shadow effect; (b) with shadow effect active



(a) (b)

Figure 4. DoF examples: (a) shallow DoF, near-to-middle distances in focus; (b) extremely shallow DoF, middle distances in focus

(a) (b)

where d is the aperture diameter, f_c the focal length, N the f-number of the lens.

The *hyperfocal distance* D_h is the distance at which the lens focus extends from half that distance to infinity. In many practical cases, the estimation of the circle of confusion for a lens can be done by setting the infinity mark on the distance scale opposite to the largest marked f-number on the DoF scale, in which situation the focus index is the hyperfocal distance. Since lens distance scales usually show object to camera distances, then if O_h is the indicated hyperfocal distance the formula should be corrected as

$$CoC \approx \frac{f_c^2}{N \cdot \left(O_h - 2f_c\right)}$$

These computations refer to how the DoF effect appears in real life lenses. In computer graphics, the virtual *pinhole camera* projects the 3D world through a zero-size lens in order to create the 2D view on a flat screen. Thus, the camera has infinite DoF and the real-life DoF effect has to be simulated. In any case the circle of confusion still has to be considered. One way to do so is by estimating CoC from the camera to object distance and the camera parameters,

$$CoC = \left| d \frac{f_c}{D_f - f_c} \frac{D_f - D_o}{D_o} \right|$$

in which d is the diameter of the lens, f_c is the focal length of the lens, D_o is the object distance and D_f the plane in focus (the distance ideally being focused on the image plane, following the fundamental formula $\frac{1}{D_f} + \frac{1}{I_f} = \frac{1}{f_c}$, I_f being the image plane to lens internal distance). The object distance can be calculated from the z-buffer z values, using the extreme far and near z-buffer values z_f, z_n,

$$D_o = -\frac{z_f z_n}{z \cdot \left(z_f - z_n\right) - z_f}$$

Alternatively, the circle of confusion may be estimated directly from the z-buffer,

$$CoC = \left|z \cdot CoC_s + CoC_b\right|, \begin{cases} CoC_s = \dfrac{d \cdot f_c \cdot D_f \cdot \left(z_f - z_n\right)}{\left(D_f - f_c\right) \cdot z_f \cdot z_n} a\, scaling\, term \\ CoC_b = \dfrac{d \cdot f_c \cdot D_f \cdot \left(z_f - z_n\right)}{D_f \cdot f_c \cdot z_n} a\, bias\, term \end{cases}$$

In order to simulate the DoF effect a method has to approximate the blur that would happen as if the lens were real. Practical implementations of the DoF effect can be categorized in the following approaches:

- Forward-mapped z-buffer techniques (Potmesil & Chakravarty, 1981)
- Distributing traced rays across the surface of a (non-pinhole) lens (Cook, Porter, & Carpenter, 1984)
- Rendering from multiple cameras (also accumulation-buffer technique) (Haeberli & Akeley, 1990)
- Rendering multiple layers (Scofield, 1994)
- Reverse-mapped z-buffer techniques (Arce & Wloka, 2002; Demers, 2003)

Antialiasing

In signal processing *antialiasing* refers to various techniques to tackle the problems of aliasing in a any sampled signal. Aliasing is an effect that causes different signals to appear the same when sampled. When aliasing occurs in signals sampled in time is referred to as *temporal aliasing*, whereas when it occurs in spatially sampled signals is called *spatial aliasing* and takes the form of *moiré patterns*. In spatial aliasing, the effect of aliasing is most apparent in situations in which the Nyquist sampling limit (sampling theorem) breaks during a sampling process (discrete sampling below the Nyquist frequency). To suppress aliasing, one may either increase the sampling frequency or suppress the high frequencies in the original signal using filtering; antialiasing methods fall in the second category of measures against aliasing. Antialiasing methods try to restrict the bandwidth of the signal so that the sampling (completely in theory or approximately for all practical purposes) satisfies the sampling theorem, over whichever band of interest is preselected in the specific application. In a simple application of computing and plotting a single white point on a black background the computations include

$$I_{\left(\hat{x}_i, \hat{y}_j\right)} = I_{\left(x,y\right)} + \left(1 - \left|x - \hat{x}_i\right|\right) \cdot \left(1 - \left|y - \hat{y}_j\right|\right)$$
$$\text{for} \begin{cases} i = \lfloor x \rfloor \ \text{ to} \lceil x \rceil \\ j = \lfloor y \rfloor \ \ to \lceil y \rceil \end{cases}$$

where $I_{\left(\hat{x}_i, \hat{y}_j\right)}$ is the computed intensity of the point at the target location $\left(\hat{x}_i, \hat{y}_j\right)$, i, j varying between the floor and ceiling of the x, y coordinates accordingly, given the initial location $\left(x, y\right)$ and intensity $I_{\left(x,y\right)}$

In modern computer graphics applications, antialiasing is usually referenced as an image effect. The antialiasing image effect is calculated in real-time, which gives smoother appearance of the graphics based on the difference of colored areas of the image (William, 1980; Franklin, 1977; Bineta, 2013; Bineta, 2013; Korein & Badler, 1983). It can be defined as a technique to minimize the distortion artifacts known as aliasing when showing a high-resolution image in a lower resolution

Figure 5. Antialised view, with some antialised shadows

display. Practically, antialiasing means removing signal components that have a higher frequency than those that can be properly resolved by the recording (or display) device (Franklin, 1977). An example of antialiasing is shown in Figure 5.

Motion Blur

Motion blur is the apparent streaking of rapidly moving objects in a still image or a sequence of images such as a movie or animation. During recording of a single image, motion blur appears as a capture of the motion of moving objects, which may occur either due to rapid movement or long exposure. Obviously this effect is the result of limitations of the recording devices to freeze-capture fast moving objects, or even slow moving objects but under long exposure requirements. Motion blur can take any of three forms, including (a) *linear blur*, which travels along a single direction to a specific extend, denoted by a strength and an angle, (b) *radial blur*, which is responsible for circular blurring effects around a center and can be defined by a radius and a strength, and (c) *zoom blur*, which represents a zooming in or out motion centered at a point on the image place defined by a center and a speed (the strength in this case).

In computer graphics there is no such problem, so in order to enhance visual realism, this effect has to be simulated. Let us consider, for example, the simple 2D case that is easy to follow, and suppose there is an image $f(x,y)$ undergoing the motion. In the duration T of the exposure, the total exposure at any point will be the integration of instantaneous exposures at that point, leading to a blurry image $g(x,y)$,

$$g(x,y) = \int_0^T f\left(x - x_0(t), y - y_0(t)\right) dt$$

where $x_0(t)$, $y_0(t)$ are the time-varying motion parameters. It is easy to see that its Fourier transform is

$$\mathcal{G}(u,v) = \iint_{-\infty}^{\infty} g(x,y) e^{-2\pi i(ux+vy)} dxdy =$$

$$= \iint_{-\infty}^{\infty} \left[\int_0^T f\left(x - x_0\left(t\right), y - y_0\left(t\right)\right) dt \right] e^{-2\pi i\left(ux+vy\right)} dxdy =$$

$$= \int_0^T \left[\iint_{-\infty}^{\infty} f\left(x - x_0\left(t\right), y - y_0\left(t\right)\right) e^{-2\pi i\left(ux+vy\right)} dxdy \right] dt =$$

$$= \int_0^T \mathcal{F}\left(u,v\right) e^{-2\pi i\left(ux_0\left(t\right)+vy_0\left(t\right)\right)} dt =$$

$$= \mathcal{F}\left(u,v\right) \int_0^T e^{-2\pi i\left(ux_0\left(t\right)+vy_0\left(t\right)\right)} dt \Rightarrow$$

$$\Rightarrow \mathcal{G}\left(u,v\right) = \mathcal{H}\left(u,v\right) \cdot \mathcal{F}\left(u,v\right)$$

If the motion parameters $x_0\left(t\right)$, $y_0\left(t\right)$ are known then $\mathcal{H}\left(u,v\right)$ is easily defined and motion readily computed. Consider the simple case in which there is a linear motion blur, in which,

$$x_0\left(t\right) = a\frac{t}{T} \qquad y_0\left(t\right) = b\frac{t}{T}$$

Then the transfer function $\mathcal{H}\left(u,v\right)$ can be defined,

$$\mathcal{H}\left(u,v\right) = \int_0^T e^{-2\pi i\left(ua\frac{t}{T}+vb\frac{t}{T}\right)} dt = \int_0^T e^{-2\pi i\frac{ua+vb}{T}t} dt =$$

$$= -\frac{T}{2\pi i\left(ua+vb\right)} e^{-2\pi i\frac{ua+vb}{T}t}\Bigg|_0^T =$$

$$-\frac{T}{2\pi i\left(ua+vb\right)} (e^{-2\pi i\left(ua+vb\right)} - 1) \Rightarrow$$

$$\Rightarrow \mathcal{H}\left(u,v\right) = \frac{T}{2\pi i\left(ua+vb\right)} \left(1 - e^{-2\pi i\left(ua+vb\right)}\right)$$

Since the transfer function is defined then $\mathcal{G}\left(u,v\right) = \mathcal{H}\left(u,v\right) \cdot \mathcal{F}\left(u,v\right)$ results the Fourier representation of the motion blurred image, which in turn gives the blurred image by taking the inverse Fourier transform.

Things get rather complicated in the 3D domain. The spatial dimensions become three, and the temporal dimension is also incorporated in more complex ways. In addition, no matter what happens in a 3D virtual world, it is the screen space 2D projection that should comply with the observers' experience. A typical mathematical formulation of motion blur in 3D graphics resolves for the 2D projected screen space $I\left(x,y\right)$,

$$I\left(x,y\right) = \sum_i \int_\Omega \int_{\Delta T} r\left(\omega,t\right) g_i\left(\omega,t\right) L_i\left(\omega,t\right) \; dt \; d\omega$$

for object i, a solid angle Ω and the virtual aperture time ΔT, in which $r\left(\omega, t\right)$ is the reconstruction filter that accounts for the effects of shutter shape and efficiency, lens aberration and film influence, $g_i\left(\omega, t\right)$ is a function of the occlusion of object i and $L_i\left(\omega, t\right)$ the radiance of the object.

The numerous rendering methods that have been proposed can be classified into (a) *analytic methods* that use a closed form solution (Korein & Badler, 1983; Grant, 1985), (b) *geometric substitution* that substitute the original geometry with alternative geometric primitives for which temporal changes are followed (Catmull, 1984; Glassner, 1988; Guan & Mueller, 2004; Jones & Keyser, 2005; Schmid, Sumner, Bowles, & Gross, 2010) (c) *texture clamping*, which originally targets the elimination of aliasing effects (Loviscach, 2005), (d) *Monte Carlo methods* that include various stochastic approaches (Haeberli & Akeley, 1990; Dayal, Watson, & Luebke, 2002; Dayal, Woolley, Watson, & Luebke, 2005; Walter, Arbree, Bala, & Greenberg, 2006; Nehab, Sander, Lawrence, Tatarchuk, & Isidoro, 2007; Kim & Ko, 2007; Akenine-Moeller, Munkberg, & Hasselgren, 2007; Hachisuka, Jarosz, Weistroffer, Dale, Humphreys, Zwicker, & Jensen, 2008; Egan, Tseng, Holzschuch, Durand, & Ramamoorthi, 2009; Overbeck, Donner, & Ramamoorthi, 2009; Ragan-Kelley, Lehtinen, Chen, Doggett, & Durand, 2010), (e) *post-processing methods* that are screen space methods that use selected pre-renders of a scene blurred by using motion information from available scene data, decoupling motion blurring from rendering (Max & Lerner, 1985; Chen & Williams, 1993; Zheng, Koestler, Thuerey, & Ruede, 2006; Vlachos, 2008; Sousa, 2008), (f) *hybrid techniques* (Sung, Pearce, & Wang, 2002) and (g) *physics-based approaches* that exploit mechanics and optics to device models for motion blur simulation (Lin & Chang, 2006; Telleen, Sullivan, Yee, Wang, Gunawardane, Collins, & Davis, 2007; Pachur, Laue, & Roefer, 2009).

Typical examples of motion blur are shown in Figure 6. Motion blur (Korein & Badler, 1983; Dachille & Kaufman, 2000; Nathan & Keyser, 2005; McGuire, Hennessy, Bukowski, & Osman, 2012) is an important method for increasing the visual quality of real-time applications such as games or training simulators.

Particle Systems

A *particle system* (Reeves, 1983) is a system of a large number of elementary graphics particles having a specific appearance, sharing a predetermined formation pattern and following specific rules, which can be used to represent fuzzy objects of processes otherwise difficult or impossible to represent by a single model. Such objects and processes include highly chaotic systems, natural phenomena or processes caused by chemical reactions, such as fire (Nguyen, Fedkiw, & Jensen, 2002), smoke (Csuri, Hackathorn, Parent, Carlson, & Howard, 1979; Huang, Gong, & Liang, 2015), explosions (Yngve, Obrien, & Hodgins, 2000), water (Foster & Fedkiw, 2001), ocean wave (Hinsinger & Neyret, 2002), clouds (Gardner, 1985), fogs, snow (Tan & Fan, 2011), dust (Jim, Xiadong, & Wegman, 1999), electricity, stars and galaxies. In essence, particle systems are used to realistically represent fluids and gasses, or in general, objects and processes with highly dynamic behavior. Instead of using geometric primitives, in these cases particle clouds are being used to render dynamically changing volumes. Stochastic processes are used to control those objects' shapes and appearance, adopting the paradigm of stochastic procedural modeling.

Particle systems are also used for modeling more tangible objects such as complicated trees (Reeves, 1985), cloths and fabric (Breen, 1994; Eberhardt, Weber, & Strasser, 1996). Realistic particle movement is often achieved by simulating real-world physics (Reynolds, 1999). The diversity of particle system applications demonstrates their importance in modern interactive media and games especially in serious games environments (Hastings, Guha, & Stanley, 2007). In recent years the researchers focus on the creation of particles APIs, in order to increase the usability and potentials of the particle systems (McAllister, 2000). In general, a particle system and its particles have very similar parameters, but with different values (Reeves, 1983): position (orientation in 3D space and center location x, y, and z), movement (velocity, rotation, etc.), color (RGB), transparency (alpha), shape (point, line, sphere, cube, rectangle, etc.), volume, density, mass, lifetime (start and end time) and blur head and rear pointers (only for particles). Figure 7 depicts a view of a 3D environment where several particle systems have been included, such as rain, snow, wind and fire.

The position, shape, and size of a particle system determine the initial positions of the particles and their range of movement. The movements of the particles are restricted within the range defined by their associated particle system. The shape of a particle system can be a point, line segment, sphere, box, or cylinder. Internal or external forces and the effects of the rotations and accelerations of the particles affect the movement of a particle system, as a whole. A particle system may change its shape, size, color, transparency, or some other attributes as it evolves. The lifetime defines how many time slices (frames) a particle will be active. A particle has both a head position and a tail position. The head position is usually animated and the tail position follows along for motion blur. The simulation looks more dynamic and has more particles with motion blur, at the cost of longer rendering time.

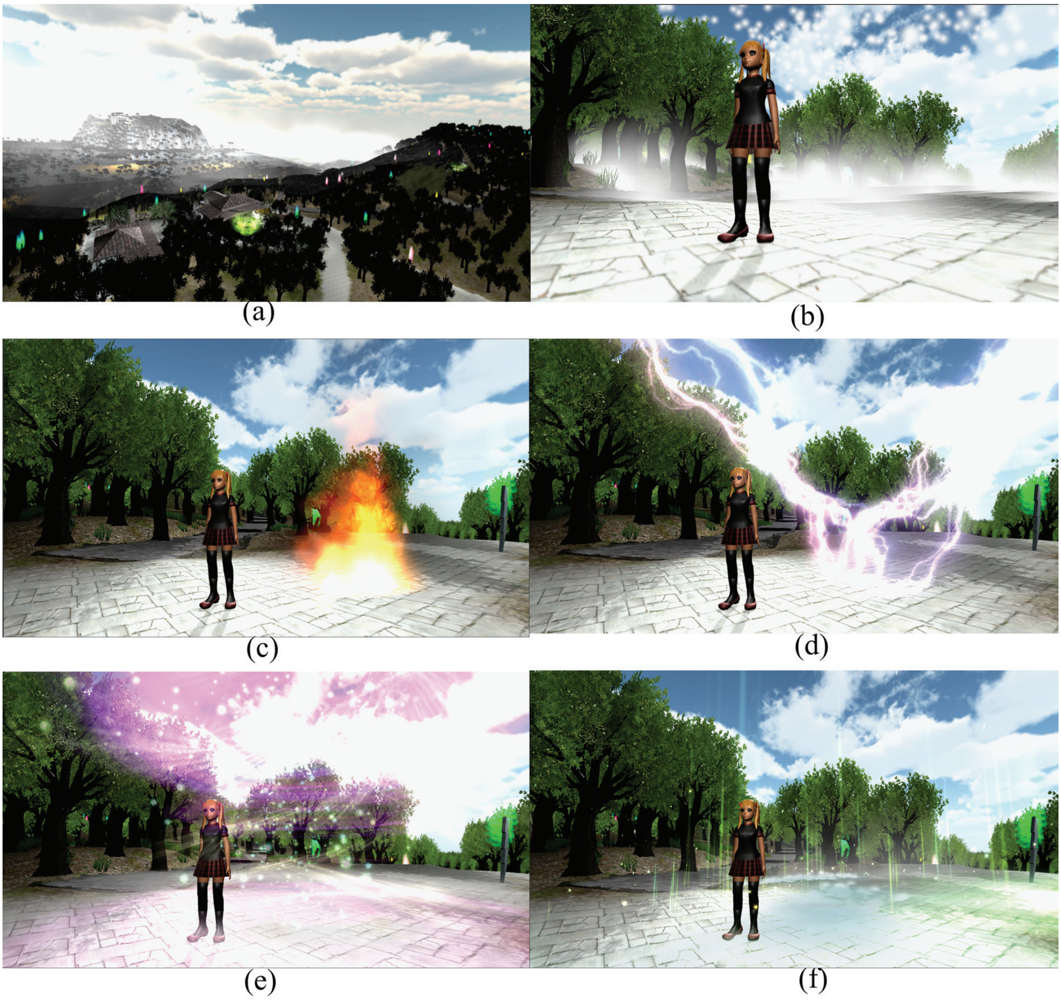
Generally, a particle systems starts by spawning particles using *emitters*, which are able to generate large numbers of elementary particles. The laws governing the particle dynamics start by defining a force field and the integration of the laws of mechanics. In a simple case, each particle is independent and considered separately (individually) but in the most general case there might be forces among the particles. In a simple example of a *sprinkler*, the mathematics can be defined easily using elementary laws of mechanics (boldface letters representing vectors),

$$\begin{aligned}
\boldsymbol{x}_{i+1} &= \boldsymbol{x}_i + d\boldsymbol{x} = \boldsymbol{x}_i + \boldsymbol{v}_i \cdot dt \\
\boldsymbol{v}_{i+1} &= \boldsymbol{v}_i - g \cdot dt \\
\boldsymbol{x}\left(t_0\right) &= \boldsymbol{x}_0 + \boldsymbol{x}_r \\
\boldsymbol{v}\left(t_0\right) &= \boldsymbol{v}_0 + \boldsymbol{v}_r
\end{aligned}$$

where there is a gravitational field g in which particles are spawned at t_0 at an initial location $\boldsymbol{x}_0$ with an initial velocity $\boldsymbol{v}_0$, both of which can be randomized by drawing random samples $\boldsymbol{x}_r$ and $\boldsymbol{v}_r$ from predefined distributions (usually seeded by uniform distributions spanning $\left[-1,1\right]$). In addition, the generation of particles can be controlled by a spawning rate r (in particles per second), which produces $n = \left(t - t_{last}\right) \cdot r$ particles, updating the t_{last} if $n > 0$. Generally, a particle is described by its state vector $\boldsymbol{X} = \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{v} \end{pmatrix}$, which is composed of six (6) numbers for a point mass particle in 3D. The dynamics of the particle can be described by the derivative of the state vector,

Figure 6. Motion blur (a) of an animated avatar and (b) of a moving scene

Figure 7. Particle systems, fog, snowfall, fire, lightning strike, cyclone and magic lights



$$\frac{d\boldsymbol{X}}{dt} = f\left(\boldsymbol{X}, t\right) = \begin{pmatrix} \boldsymbol{v} \\ \frac{\boldsymbol{F}\left(x, v\right)}{m} \end{pmatrix}, \qquad with\ \boldsymbol{F} = m \cdot \boldsymbol{a} = m\frac{d\boldsymbol{v}}{dt}$$

which generalizes for N point masses,

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{v}_1 \\ \vdots \\ \boldsymbol{x}_N \\ \boldsymbol{v}_N \end{pmatrix}_{6N\times 1}, \quad \frac{d\boldsymbol{X}}{dt} = f\left(\boldsymbol{X}, t\right) = \begin{pmatrix} \boldsymbol{v}_1 \\ \boldsymbol{F}_1 \\ \vdots \\ \boldsymbol{v}_N \\ \boldsymbol{F}_N \end{pmatrix}_{6N\times 1}$$

in which the indices represent the corresponding particle. For a single point mass m the force $\boldsymbol{F}$ is easily defined for the 3D space,

$$\boldsymbol{F} = \begin{pmatrix} 0 \\ 0 \\ -mg \end{pmatrix}$$

which can also be used in reverse direction (as buoyancy) in simulating smoke, etc. In case particles are not independent then gravity should depend on all mass interactions, taken pairwise for particles i and j at a distance or r_{ij}^2,

$$\left\| F_{ij} \right\| = \frac{G m_i m_j}{r_{ij}^2}, \quad G = 6.67 \times 10^{-11} Nm^2 \; / \; kg^2$$

in which case computations get complicated as they are $O\left(n^2\right)$.

Nevertheless, these are relatively simple cases as realistic simulations usually require the introduction of other forces, such as viscous damping or other spatial interactions such as attractive and repulsive forces, or even other dynamics such as crowd behaviors, collisions and collision avoidance.

ARTIFICIAL INTELLIGENCE IN VIRTUAL ENVIRONMENTS

One important ingredient in realistic simulations are the intelligent agents, which are the subject of major controversy over their origin, formal background, definitions, methods, applications and future directions (Ritchings, Anastassakis, & Panayiotopoulos, 2001). For social simulations to be meaningful, it is necessary to implement realistic models for both the virtual agents (CaI, Liu, Yu, & Zhang, 2015), the environment (Pelechano, Allbeck, & Badler, 2007) and their interactions; and the latter has long been the subject of multi-agent systems and reinforcment learning research (Gosavi, 2003; Varshavskaya, Kaelbling, & Rus, 2008; Silva, Melo & Veloso, 2016). A lot of attention has been given to the definition of accurate models for agents (e.g., behavioral, decision-making and interactions models) (Al-Zinati & Wenkstern, 2015). The approach of building intelligent virtual environments is relatively new in virtual environments (Pelechano, Allbeck, & Badler, 2007), in which the virtual environment is artificially being developed and created, sometimes in real time, based on the progress and the profile of the users (Al-Zinati & Wenkstern, 2015).

In addition, the realistic simulation of virtual crowds has diverse applications in architecture design, emergency evacuation, urban planning, personnel training, education, virtual museums and entertainment. The are several approaches that aim to create better representations of crowds in virtual environments, with the *agent-based methods* being the most successful; agent-based methods focus more on individual behavior, whereas crowd simulations aim to exhibit emergent phenomena of groups (Sud, Andersen, Curtis, Lin, & Manocha, 2008; Treuille, Cooper, & Popović, 2006). Not surprisingly, combinations of these approaches (which, however, do exhibit a bias in favor of one of the approaches) can deliver results of substantially enhanced realism (Shao & Terzopoulos, 2005) and, often, the crowd that is being simulated by agent-based models is designed as a multi-agent system, in order to develop different behaviors that may cooperate or compete each other (Bennewitz & Burgard, 2001; Li & Chou, 2003; Musse & Thalmann, 1997; Hanna & Richards, 2014). This approach increases the realism of the simulated crowds by highlighting the social attitude of artificial intelligence agents.

Figure 8. AI agents in virtual worlds: (a) navigation area for an AI agent; (b) AI agent-based virtual guided tour

(a) (b)

An important technical challenge in AI simulations is the path planning (Sud, Andersen, Curtis, Lin, & Manocha, 2008; Stüvel, de Goeij, van der Stappen, & Egges, 2015; Naderi, Rajamäki, & Hämäläinen, 2015; Demesure, Defoort, Bekrar, Trentesaux, & Djemaï, 2016; Champagne & Tang, 2005; Anderson, McLoughlin, Liarokapis, Petridis, & Freitas, 2010). There are several different algorithms that improve the navigation of agents in virtual environments, which often result in more realistic and detailed simulations (Sud, Andersen, Curtis, Lin, & Manocha, 2008; Naderi, Rajamäki, & Hämäläinen, 2015; Pettre, Laumond, & Thalmann, 2005). An example is shown in Figure 8a, where the blue ground is the 'walkable' area for the agent, which has to plan its route taking into account any static objects acting as obstacles. The path planning and motion planning problems have long been central to robot-based systems and AI systems in general (Luna, Lahijanian, Moll & Kavraki, 2014; Lahijanian, Maly, Fried, Kavraki, Kress-Gazit & Vardi, 2016).

Another important contribution of AI in virtual environments with cultural and educational topics (like the virtual museums) that enhances the user experience is the use of virtual agents for guided tours (Bogdanovych, Rodriguez, Simoff, & Cohen, 2009; Bickmore, Pfeifer, & Schulman, 2011; Ichiro, 2008; Sookhanaphibarn & Thawonmas, 2009; Vassos, et al., 2016). A virtual agent is a computer generated, animated, AI virtual character (usually with anthropomorphic appearance) that constitutes a real-time service representative. It leads an intelligent conversation with users, responds to their questions and performs adequate nonverbal behavior. Further enhancements may include virtual agents-guides that discuss with users and present the space (Oberlander, Karakatsiotis, Isard, & Androutsopoulos, 2008), and adaptive, personalized exhibition spaces (Bonis, Stamos, Vosinakis, Andreou, & Panayiotopoulos, 2009). An example of this concept is shown in Figure 8b, where an AI (chat-bot) avatar asks a user (also represented by an avatar in the virtual world) to explore together the environment, and provide information about specific points of interest. A related approach to enhancing the personalization experience has also focused on a more graphical exploratory search technique to allow users to navigate the results of a search query (Lin, Ahn, Brusilovsky, He, & Real, 2010).

GAME ENGINES

Since the early 1990s when really appealing computer game graphics made their first appearance, game engines started to be used as tools in many scientific fields (Harrison, 2003; Craighead, Burke, & Murphy, 2008; Breuer & Bente, 2010). Nowadays, a common approach and trend in modern virtual world applications for various domains is to adopt the technology of game development, the game engines; a simple comparison of contemporary available game engines is included in the following paragraphs.

Game engines are integrated software suites that efficiently use 3D rendering pipelines, special data-structures and speed-up techniques for visualizing texture mapped 3D objects, scenes and 3D worlds in real-time with modern graphics and interaction capacities (Harrison, 2003). The choice of game engines for building dynamic realistic virtual environments was guided by the numerous possibilities and advantages offered by modern game engines and it is not limited to the presented game engines in this study (Lepouras & Vassilakis, 2004; Trenholme & Smith, 2008; Kiourt et al., 2015a; Kiourt et al., 2015b; Kiourt et al., 2016). Game engines are being extensively tested for usability and performance due to the strict requirements of contemporary games. The manufacturers and many computer game developers provide components, algorithms, tools, guides and source code, so that end-users can efficiently create new content. An extensive user support provided by large game development communities makes development platforms even more attractive. Most of the game engine functionalities are managed and exploited via a GUI, for the convenience of developers, providing a more efficient development framework when compared to virtual environment toolkits that often require additional effort to manually provide functionalities such as user interaction (Bowman, Kruijff, LaViola, & Poupyrev, 2005; Cowan & Kapralos, 2015), avatar behaviors, collision detection and management, audio management, avatar-based virtual environment interactions, embodied autonomous agents and many other properties (Lepouras & Vassilakis, 2004; Trenholme & Smith, 2008). Almost all modern game engines provide cross-platform development, which ensures their utilization over a diversity of platforms. There is a wide selection of 3D game engines available for potential use (Lewis & Jacobson, 2008). In the following paragraphs we present the basics of some of the most popular contemporary game engines that are currently available, including *Unity*, *CryEngine* and *Unreal*. We have collected all major advantages and disadvantages regarding these game engines as discussed in (Patel et al., 2003; Craighead et al., 2008; Sillaurren & Aguirrezabal, 2012; Unity3d, 2016; Unreal, 2016; Crytek, 2016; Kiourt et al., 2016). *Unity* (Patel, White, Walczak, & Sayd, 2003; Unity3d, 2016) is a 2D/3D game engine, which offers a wide range of features and a user friendly GUI. One of its main advantages is the cross-platform integration. Each application developed in Unity version 5 (the current version) may be efficiently exported in nearly any available desktop and mobile platform (Windows, Mac OS, PS, XBOX, HTML5, Android, IOS) composing a powerful game engine. It supports 3D content derived from popular 3D applications like Blender, 3DS Max, Maya, CINEMA 4D. It offers limited 3D modeling functionalities based on primitive shapes, although it provides better landscape modeling tools. In general, every 3D model needs to be created in a third party application. However, Unity boasts a large 3D content library where a wide variety of objects can be downloaded for free or be purchased. The Unity engine provides animation mechanisms with real-time editor, particle engines with their own dedicated editor, AI Systems, dynamic lighting and shadowing, water effects, 2D game functionality, networking and multiplayer functionality, integrated physics system, terrain generation system, bitmap image effects, audio/music systems and scripting using C-sharp, JavaScript and Boo programming languages. The *CryEngine* (Craighead, Burke, & Murphy, 2008; Crytek, 2016) is a product *of Crytek Studios* and it has been used by *Ubisoft Montreal* in 2004 to develop the game *Far Cry*. The engine supports a number of features such as a real-time editor with a rich tool arsenal that enable the development of realistic games. Its functionality is similar to that of Unity with some additions such as a dedicated road and river design tool, a solid modeling designer tool and scripting in C++ or LUE programming languages. Most of its development components are managed through a sophisticated GUI that also integrates the CryEngine Sandbox world editing system. The engine supports a vast range of currently available platforms and it is continuously updated to provide export functionalities to new platforms. Licensed developers are offered with the complete source code and documentation for the whole engine and its accompanying tools. One of the primary advantages of CryEngine is the solid designer tool, by which complex meshes may be created. The *Unreal engine* (Craighead, Burke, & Murphy, 2008; Unreal, 2016) was developed by *Epic Games* in C++. The engine supports high performance rendering, advanced animation features and high-quality dynamic lighting, environmental effects, particle system, blueprint visual scripting, C++ scripting, Virtual Reality, AI avatar system, terrain builder, physically-based rendering, UI, level building, animation, visual effects, physics, networking, and cross-platform development. One of the primary advantages of the engine is its powerful particle system that can handle up to one million particles real-time visualization.

Kiourt et al. (2016) reviewed these game engines and made up an interesting and compact comparative table of pros and cons of those engines. Table 1 is an updated version of that table.

RECONSTRUCTION OF THREE-DIMENSIONAL (3D) ENVIRONMENTS

An integral part of a realistic simulation for cultural heritage applications is the environment, in the form of any cultural object, site, architectural complex, or any other form of tangible material object that conveys the story to be told. *Reconstruction* is the keyword for the creation of the 3D digital counterpart of a real-world cultural object. The reconstruction of 3D environments is based on two entirely distinctive methods: (a) *3D modeling* of objects, which is the process of developing a mathematical (graphical) representation of any 3D surface of an object via specialized software, such as Autodesk 3DS MAX (3DS MAX, 2016), Blender (Blender, 2016); (b) *3D digitization* of

Table 1. A simple three-level comparison of various features in popular game engines

Feature	Unity3D v5	CryENGINE v5	Unreal Engine v4
Friendly to User	☺	☹	☹
Components	☺	☺	☺
Object development	☹	☺	😐
GUI	☺	😐	😐
Real time FX	😐	☺	☺
Screen Space effects	😐	☺	☺
Particles	😐	😐	☺
AI	☺	☺	☺
Physics	😐	☺	☺
Animation Systems	😐	😐	😐
Programming	☺	😐	😐
OpenGL	☺	😐	😐
2D Games	☺	☹	😐
3D Formats	😐	😐	😐
Level of Detail rendering	😐	😐	😐
Visual scripting	😐	☺	☺
C#	☺	☺	☹
Tools Integration	☺	😐	😐
VR Support	☺	☺	☺
Mobile Development	☺	😐	😐
Cross-platform	☺	☹	😐
Assets	☺	☺	☺
Support	☺	☺	☺

objects, which is the process of capturing the shape and appearance of real objects using geometric and spectral measurements. In practice, in many cases those two methods are used in combination and the results are impressive (in terms of rendering realism).

3D modeling is an entirely digital process that takes places on a computer using a specialized software application. It requires high specialization in using such software, and, in particularly demanding applications – like in cultural heritage reconstruction – 3D modeling requires the participation of domain experts. *Obviously, 3D modeling is more of an artist's work, whereas 3D digitization is more of an engineer's work*. 3D digitization is a process of taking highly accurate measurements using specialized techniques and equipment. Its purpose is to get a highly accurate snapshot of reality, having cultural heritage frozen in time. In general, it has to be conducted by digitization experts, although in some cases, the automation provided makes it easier for non-experts to conduct a simple 3D digitization task. There are several techniques for the 3D digitization in cultural heritage applications, each of which with its pros and cons (Pavlidis, Koutsoudis, Arnaoutoglou et al., 2006). The process of *3D digitalization* consists of two steps, the first step is to collect the data (images, distances, sizes etc.), the second step is to analyze the collected data and develop the objects via specialized algorithms. Nowadays the most common 3D digitalization techniques include a number of methods usually falling into two main categories, namely (a) light-dependent and (b) light-independent methods. The first category includes (1) active and (2) passive methods. In summary, Table 2 presents a list of the contemporary 3D digitization methods. Pavlidis et al. (2007) summarized the various parameters involved in categorizing the cultural objects in three axes, namely the *size and shape, morphological complexity* and *diversity of materials*. This 3D space of cultural objects, graphically illustrated in Figure 9, forms a basis for a number of challenges in 3D digitization projects. Unfortunately, there is not a single 3D digitization method to fit all digitization projects. For example, the lower dark-gray region in Figure 9 corresponds to a region in which laser triangulation methods can be successful, whereas the upper medium-gray region corresponds to a region in which range-scanning methods are efficient and the central circular light-gray region depicts a region in which photogrammetry-based methods such as structure from motion are expected to be successful.

Structure from motion or SfM (Schönberger & Frahm, 2016) is one of the most widely used technique, since it is one of the easiest, most efficient and economical solutions, which does not require mastering any highly-specialized skills and equipment. A camera, a computer and a software application that can be open source, like Visual SFM (VisualSFM, 2016), can produce incredibly realistic models, but, the hardware requirements increase when the quality targeted and the size and complexity of the objects grow. Figure 10 depicts an example of a reconstruction using the SfM method for a typical cultural heritage 3D digitalization project; the blue rectangles represent the resolved camera sensor plane for each captured photo, and the small black vectors help identify the orientation and direction of the camera in the virtual 3D space, whereas the reconstructed 3D model is clearly visible.

GAMIFICATION

Developers and administrators of virtual museums have already started to realize the potential of new technologies for the development of edutainment content and services for their visitors (Bickmore, Pfeifer, & Schulman, 2011). In addition, gaming for educational purposes is a significant and active research domain (Brown & Vaughan, 2010; Nicholson, 2011; Nicholson, 2012; Pavlidis, 2015; Seaborn & Fels, 2015). This has taken either the form of game-based learning or serious gaming. *Gamification* is the result of applying game mechanics into diverse domains, in order to engage users and enhance their knowledge and performance.

The importance of playing has been emphasized in many studies from various domains. According to Brown & Vaughan (2010), playing is an archetypical activity that arises from primordial biological structures existing even before the conscience or the capacity for speech; as Brown & Vaughan emphatically stated it is not something a person decides to do.

Table 2. Summarized list of 3D digitization methods

Light-dependent methods	Active methods The light is emitted and is being detected by the digitization system	Laser triangulation
		Time-of-flight scanning (or LiDAR, LADAR, range scanning)
		Structured-light scanning
		Shape from Photometry
		Shape from Shading
		Shape from Shadow
		Tomography (of any kind)
		Speckle photography
		Holography
		Microscopy (of any kind)
	Passive methods The natural ambient light is being used by the digitization system	Photogrammetry
		Structure from Motion
		Shape from Silhouette
		Shape from Stereo
		Shape from Texture
		Shape from Focus (zooming)
Light-independent methods	Topographic methods	
	Empirical methods	
	Contact sensing methods	

According to Nicholson (2011; 2012), gamification is nothing more than the use of specific game design approaches and techniques in various environments, in order to attract people in problem solving and to enhance their contribution. Nicholson (Nicholson, 2012) shows the distinction between *Play* and *Game* using a "playful" mathematical approach,

$$Game = Play + Goals + Structure$$

This equation tries to represent a gamification definition based on the meaning of the terms *game, play, goals* and *structure*. As in any equation, the terms may be rearranged, and interesting equations result, like,

$$Game - Play = Goals + Stucture$$

$$Play = Game - \left(Goals + Structure\right)$$

In the first variant, a *game without* play is not a play-centered activity, and is largely used in gamification strategies like BLAP (Badges, Levels and Leaderboards, Achievements, and Points). A key strategy of meaningful gamification is represented in the second variant, in which by solving for *Play*, participants experience a true playful meaning in the underlying activity (Nicholson, 2012).

Figure 11 represents a cross-domain graph in an attempt to define gamification and specifically serious games, as a kind of a cross section among a number of domains formulated into three major classes that represent the theory, the content and the game design.

Figure 9. The 3D space of cultural objects

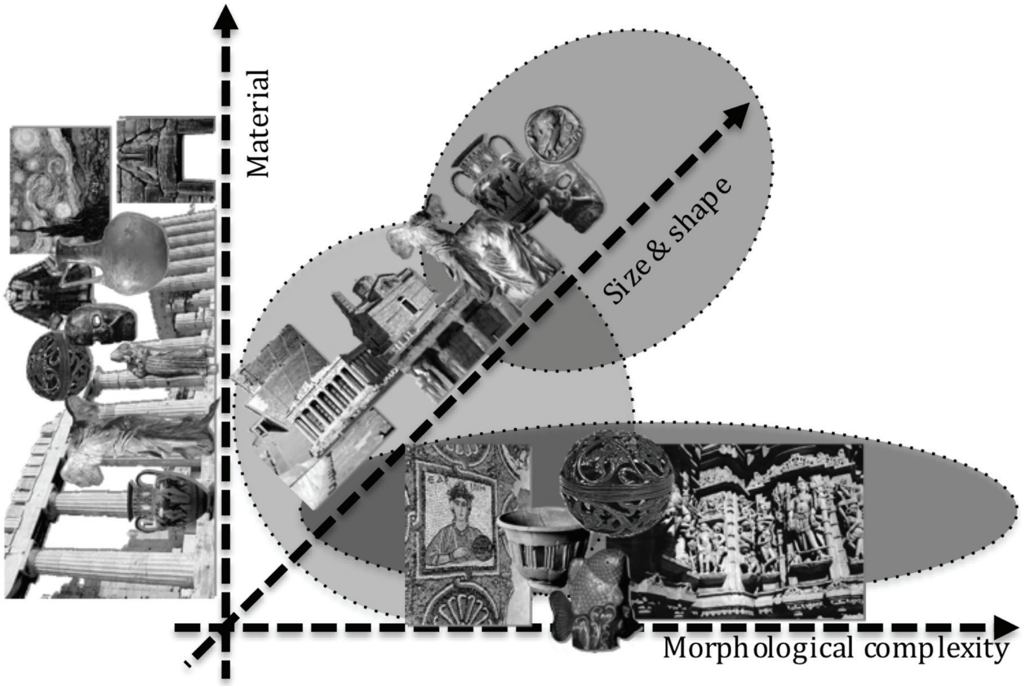
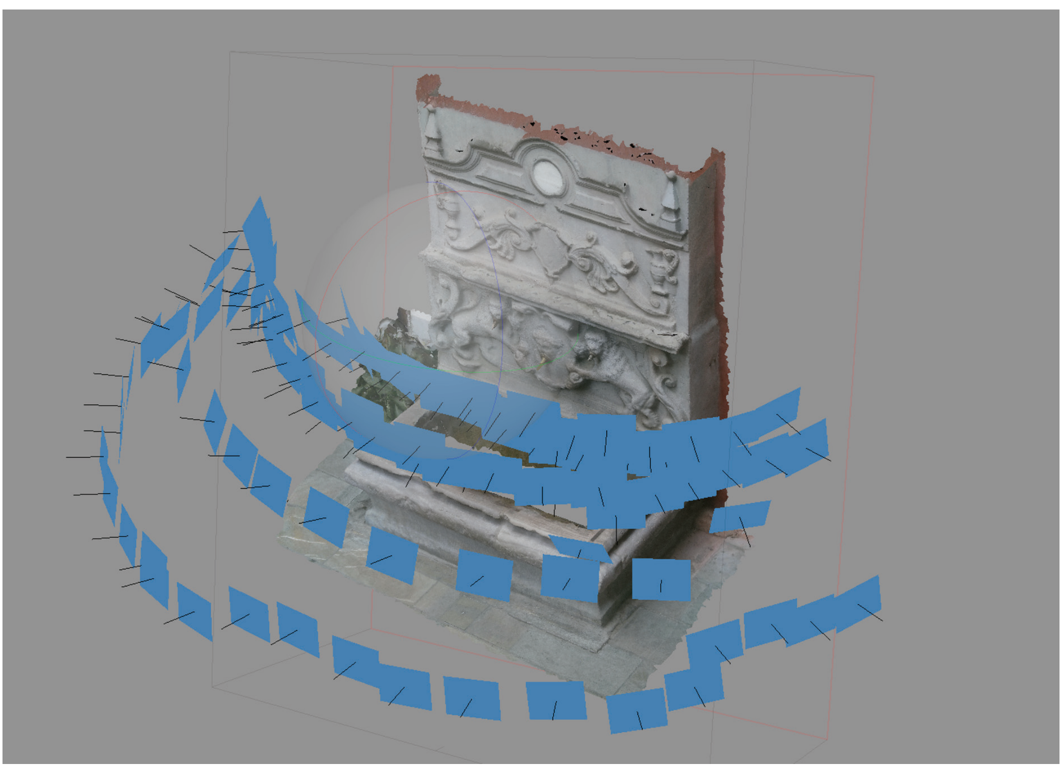


Figure 10. A typical result of the application of SfM for cultural heritage 3D digitization

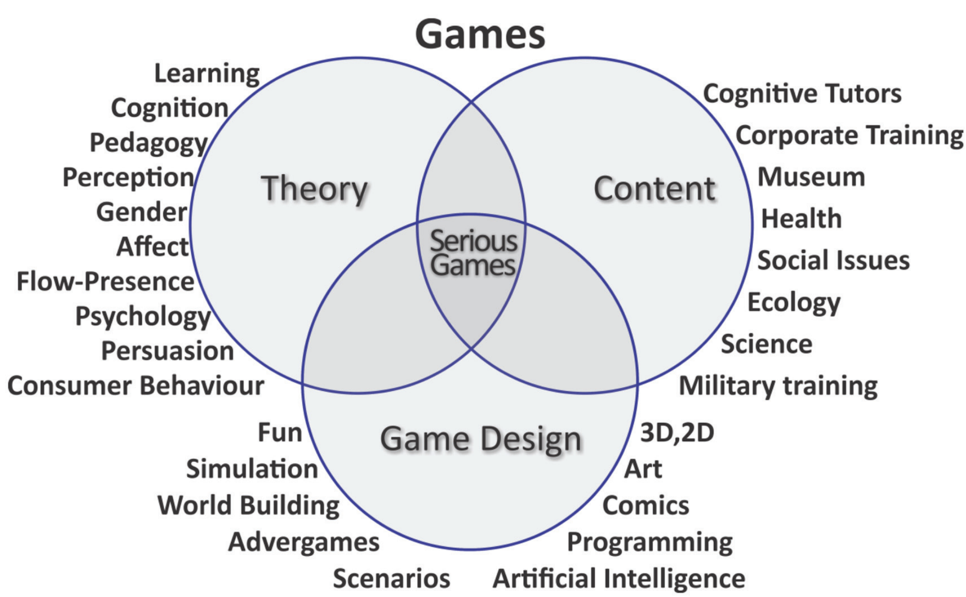


Gamification and its special application in education have a long history dating back to the 1980s. Today, with the advent of mature 3D technologies both in hardware and software implementations constitute a stable environment to successfully couple culture and education using technology and gaming as the catalysts. Various independent R&D efforts worldwide exploit this stable environment to deliver the next generation gamified virtual worlds for even more realistic and engaging cultural heritage simulations.

CONCLUSION

Realistic simulations of cultural heritage are multidimensional and a cross-disciplinary processes. In this work we made an attempt to summarize most of the key techniques, approaches and technologies available today for developing realistic 3D virtual environments for cultural end educational purposes that aim to create pleasant and engaging virtual environments for all ages. Contemporary efforts focus on various aspects including the representation of real life environments, realistic simulation of human features, interactive interfaces in the world, realistic objects and dynamic environments. In addition, interest in terms of the dynamic processes in gaming environments focuses on competitiveness and cooperation and dynamic feedback. Finally, in terms of smart virtual environments R&D efforts tackle issues in smart guided tours, smart chat-bots and smart interactive entities. Virtual environments cannot and do not intend to replace real environments, either museums or laboratories or anything else. They can be regarded as reflections of physical environments either existing or fictitious, created for various reasons and goals. There are indications that if used complementarily and as extensions of physical environments they may be most effective in enhancing engagement with culture and education with all positive outcomes for a large variety of target groups, ranging from domain experts to kids at primary school.

Figure 11. The complex multi-dimensional world of gamification in serious games



REFERENCES

Addison, A. C. (2000, April-June). Emerging Trends in Virtual Heritage. *IEEE MultiMedia*, *7*(2), 22–25. doi:10.1109/93.848421

Akenine-Moeller, T., Munkberg, J., & Hasselgren, J. (2007) Stochastic rasterization using time-continuous triangles.*Proceedings of the 22nd ACM SIGGRAPH/EUROGRAPHICS Symposium on Graphics HardwareGH '07*, San Diego, CA, USA (pp. 7–16).

Al-Zinati, M., & Wenkstern, R. (2015). A Self-Organizing Virtual Environment for Agent-Based Simulations. *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, Istanbul, Turkey (pp. 1031-1039).

Amanatides, J. (1987). Realism in Computer Graphics: A Survey. *IEEE Computer Graphics and Applications*, *7*(1), 44–56. doi:10.1109/MCG.1987.277026

Anderson, E., McLoughlin, L., Liarokapis, F., Petridis, C. P., & Freitas, S. (2010). Developing serious games for cultural heritage: A state of the art review. *Virtual Reality (Waltham Cross)*, *14*(4), 255–275. doi:10.1007/s10055-010-0177-3

Arce, T., & Matthias, W. (2002). In-Game Special Effects and Lighting. Retrieved from http://www.nvidia.com/object/gdc_in_game_special_effects.html

Autodesk, 3DS MAX. (2016). Overview. Retrieved from http://www.autodesk.com/products/3ds-max/overview

Barsky, B. A., Horn, D. R., Stanley, K. A., Jeffrey, P. A., & Yu, M. (2003). Camera models and optical systems used in computer graphics: Part I, object based techniques.*Proceedings of the 2003 International Conference on Computational Science and its Applications (ICCSA'03)* (pp. 246–255). doi:10.1007/3-540-44842-X_26

Barsky, B. A., & Kosloff, T. J. (2008). Algorithms for Rendering Depth of Field Effects in Computer Graphics. *Proceedings of the 12th WSEAS international conference on Computers* (pp. 999-1010).

Bavoil, L., & Sainz, M. (2009). Multi-layer dual-resolution screen-space ambient occlusion. Proceedings of SIGGRAPH '09, New York, NY, USA. ACM. doi:10.1145/1597990.1598035

Bellotti, F., Berta, R., Gloria, A. D., Dursi, A., & Fiore, V. (2012). A serious game model for cultural heritage. *ACM Journal on Computing and Cultural Heritage*, *5*(4), 17.

Bennewitz, M., Burgard, W., & Thrun, S. (2001). Finding solvable priority schemes for decoupled path planning techniques for teams of mobile robots. Proceedings of the 9th Int. Symposium on Intelligent Robotic Systems (SIRS). *Robotics and Autonomous Systems*, *41*(2-3), 89–99. doi:10.1016/S0921-8890(02)00256-7

Bickmore, T., Pfeifer, L., & Schulman, D. (2011). Relational Agents Improve Engagement and Learning in Science Museum Visitors. *Proceedings of the10th International Conference onIntelligent Virtual Agents*, Reykjavik, Iceland (pp. 55-67). Springer. doi:10.1007/978-3-642-23974-8_7

Bineta, M. T. (2013). Edge Based Antialiasing: Quality without Compromise. *International Journal of Engineering Science and Research Technology*, *2*(6), 1627–1631.

Blender. (2016). blender. Retrieved December 15, 2016, Online at Blender: https://www.blender.org

Bogdanovych, A., Rodriguez, J. A., Simoff, S., & Cohen, A. (2009). Virtual Agents and 3D Virtual Worlds for Preserving and Simulating Cultures. Intelligent Virtual Agents. Proceedings of the 9th International Conference, Amsterdam, Netherlands (pp. 257-271). Springer.

Bonis, B., Stamos, J., Vosinakis, S., Andreou, I., & Panayiotopoulos, T. (2009). A platform for virtual museums with personalized content. *Multimedia Tools and Applications*, *42*(2), 139–159. doi:10.1007/s11042-008-0231-2

Bonis, B., Vosinakis, S., Andreou, I., & Panayiotopoulos, T. (2013). Adaptive virtual exhibitions. *DESIDOC Journal of Library and Information Technology*, *33*(3), 183–198. doi:10.14429/djlit.33.4604

Bowman, D., Kruijff, E., LaViola, J. J., & Poupyrev, I. (2005). *3D user interfaces: theory and practise*. USA: Addison Wesley.

Breen, D., House, D. H., & Wozny, M. J. (1994). A particle based model for simulating draping behavior of woven cloth. *Textile Research Journal*, *64*(11), 663–685. doi:10.1177/004051759406401106

Breuer, J., & Bente, G. (2010). Why so serious? On the relation of serious games and learning. *Journal for Computer Game Culture*, *4*(1), 7–24.

Brown, S., & Vaughan, C. (2010). *Play: How it changes the brain, Opens the Imagination, and Invigorates the Soul*. J P Tarcher/Penguin Putnam.

Cai, L., Liu, B., Yu, J., & Zhang, J. (2015). Human behaviors modeling in multi-agent virtual environment. *Multimedia Tools and Applications*.

Callieria, M., Debevecb, P., Pairc, J., & Scopig, R. (2006). A realtime immersive application with realistic lighting: *The Parthenon. Computers & Graphics*, *30*(3), 368–376. doi:10.1016/j.cag.2006.02.015

Catmull, E. (1984) An analytic visible surface algorithm for independent pixel processing.*Proceedings of the 11th Annual Conference on Computer graphics and Interactive TechniquesSIGGRAPH '84,* New York, NY, USA (pp. 109–115). ACM. doi:10.1145/800031.808586

Champagne, J., & Tang, W. (2005). *Real-time simulation of crowds using voronoi diagrams*. EG UK Theory and Practice of Computer Graphics.

Chen, S. E., & Williams, L. (1993). View interpolation for image synthesis.*Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques SIGGRAPH '93,*New York, NY, USA (pp. 279–288). ACM. doi:10.1145/166117.166153

Cohen, M. F., & Wallace, J. R. (1993). *Radiosity and Realistic Image Synthesis*. Academic Press, Inc.

Cook, R., Porter, T., & Carpenter, L. (1984). Distributed Ray Tracing. *ACM SIGGRAPH Computer Graphics*, 18(3), 137-145. doi:10.1145/800031.808590

Cook, R. L., & Torrance, K. E. (1982). A Reflectance Model for Computer Graphics. *ACM Transactions on Graphics*, *1*(1), 7–24. doi:10.1145/357290.357293

Cowan, B., & Kapralos, B. (2015). A Survey of Frameworks and Game Engines for Serious Game Development. *Proceedings of the IEEE 14th International Conference on Advanced Learning Technologies*, Athens (pp. 662-664).

Craighead, J., Burke, J., & Murphy, R. (2008). Using the unity game engine to develop sarge: A case study. *Proceedings of theSimulation Workshop at the International Conference on Intelligent Robots and Systems*.

Crytek. (2016). Cryengine. Retrieved from http://www.crytek.com/cryengine

Csuri, C., Hackathorn, R., Parent, R., Carlson, W., & Howard, M. (1979). Towards an interactive high visual complexity animation system. *Proceedings of the 6th annual conference on Computer graphics and interactive techniques SIGGRAPH '79* (pp. 289-299). doi:10.1145/800249.807458

Dachille, F., & Kaufman, A. (2000). High-Degree Temporal Antialiasing.Proceedings of Computer Animation (pp. 49-54). doi:10.1109/CA.2000.889034

Dayal, A., Watson, B., & Luebke, D. (2002). Improving frameless rendering by focusing on change. Proceedings of the ACM SIGGRAPH 2002 Conference Abstracts and Applications SIG- GRAPH '02 (pp. 201–201). New York, NY, USA: ACM. doi:10.1145/1242073.1242214

Dayal, A., Woolley, C., Watson, B., & Luebke, D. (2005). Adaptive frameless rendering. Proceedings of the SIGGRAPH '05. doi:10.1145/1198555.1198763

DeLeon, V., & Berry, R. Jr. (2000). Bringing VR to the Desktop: Are you Game? *IEEE MultiMedia*, *7*(2), 68–72. doi:10.1109/93.848433

Demers, J. (2003). Depth of Field in the 'Toys' Demo from Ogres and Fairies: Secrets of the NVIDIA Demo Team. *Presented at GDC '03*.

Demers, J. (2004). Depth of field: A survey of techniques. In GPU Gems, R. Fernando, Ed. Addison Wesley, 23, 375–390.

Demesure, G., Defoort, M., Bekrar, A., Trentesaux, D., & Djemaï, M. (2016). Navigation Scheme with Priority-Based Scheduling of Mobile Agents: Application to AGV-Based Flexible Manufacturing System. *Journal of Intelligent & Robotic Systems*, *82*(3-4), 495–512. doi:10.1007/s10846-015-0273-4

Djaouti, D., Alvarez, J., Rampnoux, O., Charvillat, V., & Jessel, J. (2009). Serious games and cultural heritage: A case study of prehistoric caves. Proceedings of the 15th International Conference on Virtual Systems and Multimedia (pp. 221–226.). doi:10.1109/VSMM.2009.40

Doulamis, A., Liarokapis, F., Petridis, P., & Mialous, G. (2012). Serious Games for Cultural Applications. In *Intelligent Computer Graphics* (pp. 97–115). Springer.

Dutre, P., Bala, K., & Bekaert, P. (2006). *Advanced Global illumination*. AK Peters. doi:10.1201/b10632

Eberhardt, B., Weber, A., & Strasser, W. (1996). A fast, flexible, particle- system model for cloth draping. *IEEE Transactions on Computer Graphics and Applications*, *16*(5), 52–59. doi:10.1109/38.536275

Egan, K., Tseng, Y.-T., Holzschuch, N., Durand, F., & Ramamoorthi, R. (2009) Frequency analysis and sheared reconstruction for rendering motion blur. *Proceedings of the SIGGRAPH (ACM Transactions on Graphics)*. doi:10.1145/1576246.1531399

Foster, N., & Fedkiw, R. (2001). Practical animation of liquids.*Proceedings of ACM SIGGRAPH* (pp. 23–30). New York, ACM Press.

Franklin, C. C. (1977). The aliasing problem in computer-generated shaded images. *Communications of the ACM*, *20*(11), 799–805. doi:10.1145/359863.359869

Gardner, G. Y. (1985). Visual simulation of clouds. *Proceedings of the 12th annual conference on Computer graphics and interactive technique SIGGRAPH '85* (pp. 297–384). doi:10.1145/325334.325248

Glassner, A. S. (1988). Spacetime ray tracing for animation. *IEEE Computer Graphics and Applications*, *8*(2), 60–70. doi:10.1109/38.504

Gosavi, A. (2003). Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning. Springer US.

Grant, C. W. (1985). Integrated analytic spatial and temporal anti-aliasing for polyhedra in 4-space. *SIGGRAPH Computer Graphics*, *19*(3), 79–84. doi:10.1145/325165.325184

Guan, X., & Mueller, K. (2004). Point based surface rendering with motion blur. *Proceedings of theSymposium on Point Based GraphicsSPBG '04* (pp. 33-40).

Guennebaud, G., Barthe, L., & Paulin, M. (2006). Real-time soft shadow mapping by backprojection. *Proceedings of the17th Eurographics conference on Rendering Techniques* (pp. 227-234).

Hachisuka, T., Jarosz, W., Weistroffer, R. P., Dale, K., Humphreys, G., Zwicker, M., & Jensen, H. W. (2008). Multi-dimensional adaptive sampling and reconstruction for ray tracing. In SIGGRAPH '08: ACM SIGGRAPH 2008 Papers (pp. 33:1–33:10). New York, NY, USA; ACM.

Haeberli, P., & Akeley, K. (1990). The accumulation buffer: hardware support for high-quality rendering. *Proceedings of the 17th Annual Conference on Computer Graphics and Interactive TechniquesSIGGRAPH '90*(pp. 309–318). New York, NY, USA: ACM. doi:10.1145/97879.97913

Hanna, N., & Richards, D. (2014). Towards a 'Smart' Collaborative Virtual Environment and Multi-agent Approach to Designing an Intelligent Virtual Agent. In *Revised Selected and Invited Papers of the 4th International Workshop on Agent Environments for Multi-Agent Systems IV* (pp. 170-187).

Harrison, L. (2003). *Introduction to 3D Game Engine Design Using DirectX 9 and C*. Berkeley: Apress. doi:10.1007/978-1-4302-0821-1

Hasenfratz, J. M., Lapierre, M., Holzschuch, N., & Sillion, F. A.Artis GRAVIRIMAG-INRIA. (2003). A Survey of Real-time Soft Shadows Algorithms. *Computer Graphics Forum*, *22*(4), 753–774. doi:10.1111/j.1467-8659.2003.00722.x

Hastings, E., Guha, R., & Stanley, K. O. (2007). NEAT Particles: Design, Representation, and Animation of Particle System Effects.*Proceedings of the IEEE 2007 Symposium on Computational Intelligence and Games*, Honolulu, HI (pp. 154-160.). doi:10.1109/CIG.2007.368092

Hinsinger, D., & Neyret, F. (2002). Interactive animation of ocean waves. *Proceedings of ACM SIGGRAPH/Eurographics symposium on computer animation* (pp. 161-166). doi:10.1145/545261.545288

Hourcade, C., & Nicolas, A. (1985). Algorithms for antialiased cast shadows. *Computers & Graphics*, *9*(3), 259–265. doi:10.1016/0097-8493(85)90052-4

Huang, Z., Gong, G., & Liang, H. (2015). Physically-based smoke simulation for computer graphics: A survey. *Multimedia Tools and Applications*, *74*(18), 7569–7594. doi:10.1007/s11042-014-1992-4

Ichiro, S. (2008). Context-Aware Agents to Guide Visitors in Museums. *Proceedings of the8th International ConferenceIntelligent Virtual Agents* (pp. 441-455). Tokyo, Japan: Springer.

Iones, A., Krupkin, A., Sbert, M., & Zhukov, S. (2003). Fast, Realistic Lighting for Video Games. *IEEE Computer Graphics and Applications*, *23*(3), 54–64. doi:10.1109/MCG.2003.1198263

Jim, X., Xiadong, F., & Wegman, J. (1999). Real-time simulation of dust behavior generated by a fast traveling vehicle. *ACM Transactions on Modeling and Computer Simulation*, *9*(2), 81–104. doi:10.1145/333296.333366

Jones, N., & Keyser, J. Real-time geometric motion blur for a deforming polygonal mesh. *Proceedings of the Computer Graphics International 2005,* Washington, DC, USA (pp. 14–18).IEEE. doi:10.1109/CGI.2005.1500358

Jonghyun, K., Jae-Hyun, J., Changwon, J., & Byoungho, L. (2013). Real-time capturing and 3D visualization method based on integral imaging. *Optics Express*, *21*(16), 18742–18753. doi:10.1364/OE.21.018742 PMID:23938790

Kim, D., & Ko, H.-S. (2007) Eulerian motion blur. *Proceedings of theEurographics Workshop on Natural PhenomenaEG-WNP '07*, Prague, Czech Republic (pp. 39–46). ACM.

Kiourt, C., Koutsoudis, A., Arnaoutoglou, F., Petsa, G., Markantonatou, S., & Pavlidis, G. (2015a). The 'Synthesis' Virtual Museum–an open virtual exhibition creation tool. *Proceedings of theInternational Workshop on Virtual Archaeology: Museums & Cultural Tourism*. Delphi, Greece.

Kiourt, C., Koutsoudis, A., Arnaoutoglou, F., Petsa, G., Markantonatou, S., & Pavlidis, G. (2015b). *A dynamic web-based 3D virtual museum framework based on open data*. Granada, Spain: Digital Heritage. doi:10.1109/DigitalHeritage.2015.7419589

Kiourt, C., Koutsoudis, A., & Pavlidis, G. (2016). DynaMus: A fully dynamic 3D virtual museum framework. *Journal of Cultural Heritage*, *22*, 984–991. doi:10.1016/j.culher.2016.06.007

Korein, J., & Badler, N. (1983). Temporal Anti-Aliasing in Computer Generated Animation. Proceedings of Computer Graphics SIGGRAPH '83 (pp. 377–388). doi:10.1145/800059.801168

Koutsoudis, A., Makarona, C., & Pavlidis, G. (2012). Content-based navigation within virtual museums. *Journal of Advanced Computing Science and Technology*, *1*(2), 73–81. doi:10.14419/jacst.v1i2.135

Koutsoudis, A., & Pavlidis, G. (2011). Content-based search and retrieval within complex 3d scenes - an application to cultural heritage. *Proceedings of the16th International Conference on Cultural Heritage and New Technologies*.

Lahijanian, M., Maly, M. R., Fried, D., Kavraki, L. E., Kress-Gazit, H., & Vardi, M. (2016). Iterative Temporal Planning in Uncertain Environments with Partial Satisfaction Guarantees. *IEEE Transactions on Robotics*, *32*(3), 583–599. doi:10.1109/TRO.2016.2544339

Lepouras, G., & Vassilakis, C. (2004). Virtual museums for all: Employing game technology for edutainment. *Virtual Reality (Waltham Cross)*, *8*(2), 96–106. doi:10.1007/s10055-004-0141-1

Lewis, M., & Jacobson, J. (2008). Game engines in scientific research. *Communications of the ACM*, *45*(1), 27–31.

Li, T. T., & Chou, H. C. (2003). Motion planning for a crowd of robots. *Proceedings of theIEEE Int. Conf. on Robotics and Automation*.

Lin, H., & Chang, C. (2006) Photo-consistent motion blur modeling for realistic image synthesis. *Proceedings of thePacific-Rim Symposium on Image and Video Technology* (pp. 1273–1282). doi:10.1007/11949534_129

Lin, Y., Ahn, J.-w., Brusilovsky, P., He, D., & Real, W. (2010). ImageSieve: Exploratory Search of Museum Archives with Named Entity-Based Faceted Browsing. In C. Marshall, E. Toms and A. Grove (Eds.), *Proceedings of 2010 Annual Meeting of The American Society for Information Science & Technology (ASIS&T 2010)*, Pittsburgh, PA.

Loviscach, J. (2005) Motion blur for textures by means of anisotropic filtering. *Proceedings of theEurographics Symposium on Rendering* (pp. 7–14).

Luna, R., & Lahijanian, M., Moll, & M., Kavraki L.E. (2014). Optimal and Efficient Stochastic Motion Planning in Partially-Known Environments. *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, Quebec City, Canada (pp. 2549–2555).

Max, N. L., & Lerner, D. M. (1985). A two-and-a-half-d motion-blur algorithm. *SIGGRAPH Computer Graphics.*, *19*(3), 85–93. doi:10.1145/325165.325188

McAllister, D. K. (2000). *The Design of an API for Particle Systems*. Chapel Hill, NC, USA: University of North Carolina at Chapel Hill, Department of Computer Science.

McGuire, M., Hennessy, P., Bukowski, M., & Osman, B. (2012). A reconstruction filter for plausible motion blur. *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, Costa Mesa, CA (pp. 135-142). doi:10.1145/2159616.2159639

Miller, G. (1994). Efficient algorithms for local and global accessibility shading. *Proceedings of the 21st annual conference on Computer graphics and interactive techniques SIGGRAPH '94* (pp. 319-326). doi:10.1145/192161.192244

Musse, S. R., & Thalmann, D. (1997). A model of human crowd behavior: Group inter-relationship and collision detection analysis. In *Computer Animation and Simulation* (pp. 39-51).

Naderi, K., Rajamäki, J., & Hämäläinen, P. (2015). RT-RRT*: a real-time path planning algorithm based on RRT*.*Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*, Paris, France (pp. 113-118). doi:10.1145/2822013.2822036

Nathan, J., & Keyser, J. (2005). Real-Time Geometric Motion Blur for a Deforming Polygonal Mesh.Proceeding of Computer Graphics International, Stony Brook, NY, USA (pp. 26-31).

Nehab, D., Sander, P. V., Lawrence, J., Tatarchuk, N., & Isidoro, J. R. (2007) Accelerating real-time shading with reverse reprojection caching.*Proceedings of the 22nd ACM SIGGRAPH/EUROGRAPHICS Symposium on Graphics HardwareGH '07,* Aire-la-Ville, Switzerland (pp. 25–35).

Nguyen, D. Q., Fedkiw, R., & Jensen, H. W. (2002). Physically based modeling and animation of fire. *Transactions on Graphics*, *21*(3), 721–728.

Nicholson, S. (2011). Making the Gameplay Matter: Designing Modern Educational Tabletop Games. *Knowledge Quest*, *40*(1), 60–65.

Nicholson, S. (2012). Strategies for meaningful gamification: Concepts behind transformative play and participatory museums. Presented at Meaningful Play 2012, Lansing, Michigan.

Oberlander, J., Karakatsiotis, G., Isard, A., & Androutsopoulos, A. (2008). Building an adaptive museum gallery in Second Life. *Proceedings of Museums and the Web 2008*, Montreal, Quebec, Canada.

Overbeck, R. S., Donner, C., & Ramamoorthi, R. (2009). Adaptive Wavelet Rendering. *ACM Transactions on Graphics*, *28*(5), 140. doi:10.1145/1618452.1618486

Pachur, D., Laue, T., & Roefer, T. (2009). *Real-Time Simulation of Motion-Based Camera Disturbances. Proceedings of RoboCup 2008: Robot Soccer World Cup XII* (pp. 591–601). Berlin, Heidelberg: Springer-Verlag. doi:10.1007/978-3-642-02921-9_51

Papagiannakis, G., L'Hoste, G., Foni, A., & Thalmann, N. M. (2001). Real-time photo realistic simulation of complex heritage edifices.*Proceedings Seventh International Conference on Virtual Systems and Multimedia* (pp. 218-227). doi:10.1109/VSMM.2001.969674

Papastamatiou, N., Alexandridis, T., Tsergoulas, K., Michopoulos, A., & Karadimas, N. V. (2010). Virtual reality applications with user interface for dynamic content development. *Proceedings of the 4th WSEAS international conference on Computer engineering and applications CEA'10* (pp. 201–206).

Patel, M., White, M., Walczak, K., & Sayd, P. (2003). Digitisation to presentation: Building virtual museum exhibitions. In *Vision, Video and Graphics*.

Pavlidis, G. (2015). Towards a playful education. *Proceedings of theInternational Workshop on Virtual Archaeology: Museums & Cultural Tourism*, Delphi, Greece.

Pavlidis, G., Koutsoudis, A., Arnaoutoglou, F., Tsioukas, V., & Chamzas, C. (2007). Methods for 3D digitization of Cultural Heritage. *Journal of Cultural Heritage*, *8*(1), 93–98. doi:10.1016/j.culher.2006.10.007

Pavlidis, G., Makarona, C., Arnaoutoglou, F., Koutsoudis, A., & Chamzas, C. (2008). Museums In The Virtual World. *Proceedings of theInternational Symposium Information and Communication Technologies in Cultural Heritage*, Ioannina, Greece (pp. 16-18).

Pavlidis, G., Tsiafakis, N., Provopoulos, G., Chatzopolos, S., Arnaoutoglou, F., & Chamzas, C. (2006, June 5-9). Momi: A dynamic and internet-based 3d virtual museum of musical instruments. *Proceedings of theThird International Conference of Museology*.

Pelechano, N., Allbeck, J., & Badler, N. (2007). Controlling individual agents in high-density crowd simulation. *Proceedings of the ACM SIGGRAPH/Eurographics symposium on Computer animation*, San Diego, California, USA (pp. 99–108).

Pettre, J., Laumond, J. P., & Thalmann, D. (2005). A navigation graph for real-time crowd animation on multilayered and uneven terrain. *Proceedings of theFirst International Workshop on Crowd Simulation*.

Phong, T. B. (1973). *Illumination for computer generated images* [PhD. Dissertation]. University of Utah, Salt Lake City, Utah, USA.

Potmesil, M., & Indranil, C. (1981). A Lens and Aperture Camera Model for Synthetic Image Generation. *Proceedings of the 8th Annual Conference on Computer Graphics and Interactive Techniques* (pp. 297–305). doi:10.1145/800224.806818

Potmesil, M., & Indranil, C. (1982). Synthetic image generation with a lens and aperture camera model. *ACM Transactions on Graphics*, *1*(2), 85–108. doi:10.1145/357299.357300

Prem, K., Thalmann, N. M., Moccozet, L., Sannier, G., Aubel, A., & Thalmann, D. (1998). Real-Time Animation of Realistic Virtual Humans. *IEEE Computer Graphics and Applications*, *18*(5), 42–56. doi:10.1109/38.708560

Ragan-Kelley, J., Lehtinen, J., Chen, J., Doggett, M., & Durand, F. (2010). Decoupled Sampling for Real- Time Graphics Pipelines (tech. report). Massachusetts Institute of Technology.

Reeves, W. T. (1983). Particle Systems — a Technique for Modeling a Class of Fuzzy Objects. *ACM Transactions on Graphics*, *2*(2), 91–108. doi:10.1145/357318.357320

Reeves, W. T. (1985). Approximate and probabilistic algorithms for shading and rendering structured particle systems. *Proceedings of the 12th annual conference on Computer graphics and interactive techniques SIGGRAPH '85* (pp. 313-322). doi:10.1145/325334.325250

Remondino, F., & El-Hakim, S. (2006). Image-based 3D Modelling: A Review. *The Photogrammetric Record*, *21*(115), 269–291. doi:10.1111/j.1477-9730.2006.00383.x

Reynolds, C. (1999). Steering behaviors of autonomous characters.*Proceedings of Game Developers Conference*, San Francisco, CA (pp. 763-782).

Ritchings, T., Anastassakis, G., & Panayiotopoulos, T. (2001). Multi-agent Systems as Intelligent Virtual Environments. *Proceedings of the Advances in Artificial Intelligence: Joint German/Austrian Conference on AI (KI '01)* (pp. 381-395).

Rogers, A. (1996). *Teaching Adults* (2nd ed.). Buckingham: Open University Press.

Schmid, J., Sumner, R., Bowles, H., & Gross, M. (2010). Programmable motion effects. *ACM Transactions on Graphics,* 29(3). doi:10.1145/1833349.1778794

Schönberger, J. L., & Frahm, J. M. (2016). Structure-from-Motion Revisited.*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Scofield, C. (1994). 21/2-D Depth of Field Simulation for Computer Animation. In D. Kirk (Ed.), *Graphics Gems III* (pp. 36–38). Morgan Kaufmann.

Seaborn, K., & Fels, D. I. (2015). Gamification in theory and action: A survey. *International Journal of Human-Computer Studies*, *87*, 14–31. doi:10.1016/j.ijhcs.2014.09.006

Selgrad, K., Reintges, C., Penk, D., Wagner, P., & Stamminger, M. (2015). Real-time depth of field using multi-layer filtering. *Proceedings of the 19th Symposium on Interactive 3D Graphics and Games,* San Francisco, CA (pp. 121-127). doi:10.1145/2699276.2699288

Shao, W., & Terzopoulos, D. (2005). Autonomous Pedestrians.*Proceedings of SIGGRAPH/EG Symposium on Computer Animation (SCA'05)*, Los Angeles, CA.

Sillaurren, S., & Aguirrezabal, P. (2012). 3d publish: A web-based solution for building dynamic 3d virtual museums. *International Journal of Heritage in the Digital Era*, *1*(1_suppl1 Suppl.), 147–152. doi:10.1260/2047-4970.1.0.147

Silva, R., Melo, F., & Veloso, M. (2016). Adaptive Symbiotic Collaboration for Targeted Complex Manipulation Tasks. *Proceedings of ECAI'16, the 22nd European Conference on Artificial Intelligence*, The Hague, Holland (pp. 863-870).

Sookhanaphibarn, K., & Thawonmas, R. (2009). A content management system for user-driven museums in second life. *Proceedings of the International Conference on CyberWorlds* (pp. 185–189). doi:10.1109/CW.2009.58

Sousa, T. (2008) Crysis next-gen effects.*Proceedings of the Game Developers Conference 2008*.

Stüvel, S. A., de Goeij, M. F., van der Stappen, A. F., & Egges, A. (2015). An analysis of manoeuvring in dense crowds.*Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*, Paris, France (pp. 85-90). doi:10.1145/2822013.2822037

Sud, A., Andersen, E., Curtis, S., Lin, M., & Manocha, D. (2008). Real-time path planning for virtual agents in dynamic environments.Proceedings of the ACM SIGGRAPH 2008 classes, Los Angeles, CA. doi:10.1145/1401132.1401206

Sung, K., Pearce, A., & Wang, C. (2002). Spatial-temporal antialiasing. *IEEE Transactions on Visualization and Computer Graphics*, *8*(2), 144–153. doi:10.1109/2945.998667

Sylaiou, S., Liarokapis, F., Kotsakis, K., & Patias, P. (2009). Virtual museums, a survey and some issues for consideration. *Journal of Cultural Heritage*, *4*, 520–528.

Sylaiou, S., Mania, K., Karoulis, A., & White, M. (2010). Exploring the relationship between presence and enjoyment in a virtual museum. *International Journal of Human-Computer Studies*, *68*(5), 243–253. doi:10.1016/j.ijhcs.2009.11.002

Sylaiou, S., Mania, K., Paliokas, I., Pujol, L., Kilintzis, V., & Liarokapis, F. (in press). Exploring the educational impact of diverse technologies in online Virtual Museums. *International Journal of Arts and Technology.*

Tan, j., & Fan, X. (2011). Particle System Based Snow Simulating in Real Time. *Proceedings of the 3rd International Conference on Environmental Science and Information Application Technology ESIAT 2011* (pp. 1244-1249).

Telleen, J., Sullivan, A., Yee, J., Wang, O., Gunawardane, P., Collins, I., & Davis, J. (2007). Synthetic shutter speed imaging. *Computer Graphics Forum*, *26*(3), 591–598. doi:10.1111/j.1467-8659.2007.01082.x

Trenholme, D., & Smith, S. P. (2008). Computer game engines for developing first-person virtual environments. *Virtual Reality (Waltham Cross)*, *12*(3), 181–187. doi:10.1007/s10055-008-0092-z

Treuille, A., Cooper, S., & Popović, Z. (2006). Continuum crowds. *ACM Transactions on Graphics*, *25*(3), 1160–1168. doi:10.1145/1141911.1142008

Tsichritzis, D., & Gibbs, S. (1991). Virtual museums and virtual realities. *Proceedings of theInternational Conference on Hypermedia and Interactivity in Museums* (pp. 17-25).

Unity3d. (2016). Unity Technologies. Retrieved from http://unity3d.com/

Unreal Engine. (2016). Retrieved from https://www.unrealengine.com/

Van Eck, R. (2006). Digital game-based learning: It's not just the digital natives who are restless. *EDUCAUSE*, *41*(2), 16–30.

Vardis, K., Papaioannou, G., & Gaitatzes, A. (2013). Multi-view Ambient Occlusion with Importance Sampling. *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, Orlando, FL, USA (pp. 111-118). doi:10.1145/2448196.2448214

Varshavskaya, P., Kaelbling, L. P., & Rus, D. (2008). Efficient Distributed Reinforcement Learning Through Agreement.*Proceedings of the 9th International Symposium on Distributed Autonomous Robotic Systems (DARS)*, Tsukuba Japan.

Vassos, S., Malliaraki, E., dal Falco, F., Di Maggio, J., Massimetti, M., Nocentini, M. G., & Testa, A. (2016). Art-bots: Toward Chat-based Conversational Experiences in Museums. *Proceedings of the9th International Conference on Interactive Digital Storytelling* (pp. 433-437). Los Angeles, CA, USA: Springer International Publishing. doi:10.1007/978-3-319-48279-8_43

Verbeck, P. C., & Greenberg, P. D. (1984). A comprehensive light source description for computer graphics. *IEEE Computer Graphics and Applications*, *4*(7), 66–75. doi:10.1109/MCG.1984.275906

Visual SFM. (2016). Retrieved from http://ccwu.me/vsfm/

Vlachos, A. (2008). Advanced Visual Effects with Direct3D: Post Processing in The Orange Box.*Proceedings of the Game Developer's Conference 2008*, San Francisco, CA.

Walter, B., Arbree, A., Bala, K., & Greenberg, D. P. (2006). Multidimensional lightcuts. Proceedings of SIGGRAPH '06 (pp. 1081–1088). ACM. doi:10.1145/1179352.1141997

William, L. L. (1980). Human Vision, Anti-aliasing, and the Cheap 4000 Line Display. *Computer Graphics*, *14*(3), 308–313. doi:10.1145/965105.807509

Wojciechowski, R., Walczak, K., White, M., & Cellary, W. (2004). Building virtual and augmented reality museum exhibitions. *Proceedings of the ninth international conference on 3D Web technology* (pp. 135-144). doi:10.1145/985040.985060

Woo, A., Poulin, P., & Fournier, A. (1990). A Survey of Shadow Algorithms. *IEEE Computer Graphics and Applications*, *10*(6), 13–32. doi:10.1109/38.62693

Yang, L., Dong, H., Alelaiwi, A., & El Saddik, A. (2015). See in 3D: State of the art of 3D display technologies. *Multimedia Tools and Applications*, *75*(24), 17121–17155.

Yngve, G. D., Obrien, J. F., & Hodgins, J. K. (2000). Animation explosions.*Proceedings of ACM SIGGRAPH 2000 conference proceedings* (pp. 29–36). ACM Press.

Yu, I., Mortensen, J., Khanna, P., Spanlang, B., & Slater, M. (2012). Visual realism enhances realistic response in an immersive virtual environment. *IEEE Computer Graphics and Applications*, *32*(6), 36–45. doi:10.1109/MCG.2012.121 PMID:24807308

Yu, X., Wang, R., & Yu, J. (2010). Real-time depth of field rendering via dynamic light field generation and filtering. *Computer Graphics Forum*, *39*(7), 2099–2107. doi:10.1111/j.1467-8659.2010.01797.x

Zheng, Y., Koestler, H., Thuerey, N., & Ruede, U. (2006). Enhanced motion blur calculation with optical flow. *Proceedings of Vision* (pp. 253–260). Modeling and Visualization.

Chairi Kiourt is a PhD Candidate at the Hellenic Open University, School of Science and Technology. He received a Bachelor degree in electrical engineering from the Technical Education Institute of Kavala, Greece. He also received an MSc in Systems Engineering and Management from the University of Thrace, Greece. His research is on large scale agent socialization experiments, using game playing and machine learning.

George Pavlidis received his PhD in Electrical Engineering, earning the distinction of the Ericsson Awards of Excellence in Telecommunications. He has been working for numerous R&D projects with emphasis on multimedia systems in culture and education. In 2002 he joined the 'Athena' Research Center, where he is now a research director, head of the Multimedia Research Group and head of research at 'Clepsydra' Cultural Heritage Digitization Center. His research interests involve 2D/3D imaging, CBIR, multimedia technologies, human-computer interaction, intelligent user interfaces, multi-sensory environments and ambient intelligence, 3D digitization and reconstruction, 3D-GIS and mixed/augmented/virtual reality. Dr. Pavlidis is a member of the Technical Chamber of Greece, of the Hellenic Researchers' Association, a senior member of the IEEE, and a founding member of the 'Athena' Research Center's Researchers' Association.

Anestis Koutsoudis has been working in the area of 3D computer graphics as a 3D graphics designer and as a software engineer since his undergraduate studies. Nowadays, he basically thinks on three dimensional scanning, real time 3D graphics programming, photogrammetry, 3D content based retrieval, 3D scene metadata annotation and inevitably software engineering. He works as a researcher in the ATHENA - Research and Innovation Centre in Information, Communication and Knowledge Technologies, where he holds an associate research fellow (B') position. He has participated in many research and development projects with emphasis on real time 3D graphics programming, 3D digitisation, content-based retrieval, multimedia and IT application in the cultural heritage domain.

Dimitris Kalles is an Associate Professor on Artificial Intelligence with the Hellenic Open University (HOU). He holds a Diploma in Computer Engineering and Informatics from the University of Patras, Greece (1991). He also holds an MSc (1993) and a PhD (1995) in Computation from the University of Manchester Institute of Science and Technology (UMIST, today University of Manchester). While with HOU, he has been involved in the design and implementation of study programs and of several projects on the development of educational material and in the implementation of infrastructures for e-learning. He has taught several subjects at Informatics at undergraduate and postgraduate levels and has supervised over 40 undergraduate and postgraduate dissertations and theses, at HOU, at the Open University of Cyprus and at University of Patras (Greece). He has worked as an R&D engineer and researcher with Research Academic Computer Technology Institute (Patras, Greece, 1997-2000), as a Technical Director with Ahead Relationship Mediators SA (Athens, 2000-2004, also serving as member of its BoD during 2000-2008). He has served as a project evaluator for the European Commission, the Corallia Clusters Initiative, and the University of Porto, and as a proposal evaluator for the European Commission, for the General Secretariat for Research and Technology and for the Ministry of Education (Greece), and for other organisations. He was a member of the Organising Committee of the 18th European Conference on Artificial Intelligence, held in July 2008 in Patras, and of the 24th International Conference on Tools with Artificial Intelligence, to be held in November 2012, in Piraeus. He co-chaired the 8th Hellenic Conference on Artificial Intelligence, held in May 2014, in Ioannina. He has served as Chairperson (2012-14) and Secretary General (2004-8, 2010-12) of the Hellenic Society for Artificial Intelligence. He is actively researching artificial intelligence and e-learning technologies and has published over 60 papers in scientific journals and conference proceedings and his work has received more than 500 citations. He is a skilled programmer.

Call for Articles

International Journal of Computational Methods in Heritage Science

Volume 1 • Issue 1 • January-June 2017 • ISSN: 2473-5345 • eISSN: 2473-5337

An official publication of the Information Resources Management Association

MISSION

The **International Journal of Computational Methods in Heritage Science (IJCMHS)** aims to present innovative research on digital solutions and advancements in the cross-disciplinary field of cultural heritage research and documentation. Through a presentation of applied and theoretical research and case studies from around the world, IJCMHS intends to be a pivotal resource for anthropologists, archaeologists, museum curators, librarians, archivists, historians as well as academics and students studying topics relevant to cultural heritage and conservation.

COVERAGE AND MAJOR TOPICS

The topics of interest in this journal include, but are not limited to:

Cultural Informatics • Digital Archaeometry • Digital Reconstruction • Documentation • Heritage Diagnostics • Heritage Materials • Historic Metallurgy • Open Data for Cultural Heritage • Public Engagement • Sustainable Heritage • Virtual Museums and Exhibitions

ALL INQUIRIES REGARDING IJCMHS SHOULD BE DIRECTED TO THE ATTENTION OF:

George P. Pavlidis, Editor-in-Chief • IJCMHS@igi-global.com

ALL MANUSCRIPT SUBMISSIONS TO IJCMHS SHOULD BE SENT THROUGH THE ONLINE SUBMISSION SYSTEM:

http://www.igi-global.com/authorseditors/titlesubmission/newproject.aspx

IDEAS FOR SPECIAL THEME ISSUES MAY BE SUBMITTED TO THE EDITOR(S)-IN-CHIEF

PLEASE RECOMMEND THIS PUBLICATION TO YOUR LIBRARIAN

For a convenient easy-to-use library recommendation form, please visit:
http://www.igi-global.com/IJCMHS