entropy

MDPI

Article

Viewpoint Selection for 3D-Games with f-Divergences

Micaela Y. Martin [1,*], Mateu Sbert [2] and Miguel Chover [1,*]

[1] Institute of New Image Technologies, Universitat Jaume I, 12071 Castellón, Spain
[2] Institute of Informatics and Applications, University of Girona, 17071 Girona, Spain; mateu.sbert@gmail.com
* Correspondence: micmarti@uji.es (M.Y.M.); chover@uji.es (M.C.)

Abstract: In this paper, we present a novel approach for the optimal camera selection in video games. The new approach explores the use of information theoretic metrics f-divergences, to measure the correlation between the objects as viewed in camera frustum and the ideal or target view. The f-divergences considered are the Kullback–Leibler divergence or relative entropy, the total variation and the χ^2 divergence. Shannon entropy is also used for comparison purposes. The visibility is measured using the differential form factors from the camera to objects and is computed by casting rays with importance sampling Monte Carlo. Our method allows a very fast dynamic selection of the best viewpoints, which can take into account changes in the scene, in the ideal or target view, and in the objectives of the game. Our prototype is implemented in Unity engine, and our results show an efficient selection of the camera and an improved visual quality. The most discriminating results are obtained with the use of Kullback–Leibler divergence.

Keywords: viewpoint selection; video games; Kullback–Leibler divergence; χ^2 divergence; total variation

1. Introduction

In the context of 3D virtual scenes in video games, the selection of the best camera position and orientation has not yet received enough attention. Considering the critical importance of visual perception in the development of the game storyline, it is fundamental to develop methods for selecting optimal views that emphasize the most important scene information for the player.

Information theory measures, mainly Shannon entropy, have been widely used in viewpoint selection in robotics, computer graphics and visualization [1,2]. A drawback of these measures is the high associated cost due to their computation with projections.

In this paper we propose the use of f-divergences, exploring the Kullback–Leibler, the total variation, and the χ^2 divergence to compute the best viewpoint or camera position in a Unity [3] environment and compare it with the Shannon entropy.

The Kullback–Leibler (K-L) divergence [4] has been introduced as a measure of the best viewpoint of an object in [5,6] and we extend it here to select the best camera in 3D scenes. To measure the visibility of an object, we introduce the camera frustum form factor, extending the work in [7,8]. The form factors of all objects plus the background form factor is a probability distribution that is compared using the K-L divergence with a target distribution, which can be the distribution of relative areas or be weighted with importance values. We also extend the viewpoint entropy to frustum entropy too and show that viewpoint frustum entropy happens when all objects are given importance inversely proportional to their area.

We use the Unity game engine as a development tool because it is one of the most popular and widely used and widely used game engines in the industry, and it offers out-of-the-box ray casting that allows us to compute the form factors in a fast and efficient way.

In this paper we advance on the state of the art along the following novelties:

check for updates

Citation: Martin, M.Y.; Sbert, M.; Chover, M. Viewpoint Selection for 3D-Games with f-Divergences. *Entropy* **2024**, *26*, 464. https://doi.org/10.3390/e26060464

Academic Editor: Udo Von Toussaint

Received: 8 December 2023
Revised: 20 May 2024
Accepted: 24 May 2024
Published: 29 May 2024

- We use the f-divergences, and in particular Kullback–Leibler divergence, total variation, and χ^2-divergence as a measure of viewpoint in a scene consisting of 3D objects, extending from the use of K-L divergence as a viewpoint measure [5].
- We define the frustum form-factor as a measure of the visibility of an object from a camera, extending the classic form-factor concept used in radiative heat transfer [9], radiosity and global illumination [10–12].
- We compute the frustum form factor with a Monte Carlo technique using the built-in ray tracing Unity routines. This allows for smooth computing and integrating the view-point measures in run-time.
- We define a target distribution that can be fine-tuned according to the importance assigned to each object and is extended with a wildcard, the background value which allows to regulate how much background should be visible from the camera.
- The frustum form-factor distribution is then compared, using the f-divergences, with the target distribution.

The rest of the paper is organized as follows. In Section 2 we review the state of the art on viewpoint measures used in robotics and computer graphics and visualization, as well as f-divergences. In Section 3 we present our framework together with the viewpoint divergence measures. Section 4 contains an evaluation of our framework, in Section 5 we discuss the results and in Section 6 we present our conclusions, limitations of our method and future work.

2. State of the Art

2.1. Viewpoint Selection

The selection of the best point of view in 3D models has been widely investigated in the scientific literature. Plemenos et al. [13] proposed to use projected area and number of polygons seen as the best viewpoint measure. Vazquez et al. [14] proposed the concept of the best viewpoint as the one that maximizes the entropy. The viewpoint entropy measure has proven to be effective in the selection of optimal viewpoints in several applications dealing with 3D environments [15], including robotics [16], and in volumetric data [17].

Bonaventura et al. [6] have proposed a comprehensive classification of attributes, such as area, silhouette, depth, stability and surface curvature to evaluate the quality of a point of view in polygonal models. Some measures included in this category are the number of visible triangles [13], projected area, visibility ratio, viewpoint entropy and Kullback–Leibler viewpoint measure [5]. Silhouette attributes focus on the shape and structure of the object visible from the point of view. Related measures are the length of the silhouette [18], the entropy of the silhouette, the curvature of the silhouette and the extreme of the curvature of the silhouette [19]. Depth attributes focus on the depth of information. Measures in this category are the measure of Stoev and Straßer [20], the maximum depth and the depth distribution. Surface curvature attributes are based on the analysis of the curvature of the surface of the visible object. The stability of the viewpoint is another important aspect to consider when selecting the best viewpoint. Stability attributes evaluate the consistency and continuity between nearby viewpoints. Related measures are instability [21], which is based on the Jensen–Shannon divergence between the projected area distributions and the areas, and visual stability based on depth [22], which uses the normalized compression distance between the depth images of viewpoints.

An important aspect pointed out by Zeng et al. [16] in viewpoint evaluation is the influence of factors such as the occlusion between objects, the different lighting configurations, materials, and textures. These factors can significantly affect the quality of the information collected and, therefore, the choice of the best point of view. Although we acknowledge that this information can be very valuable in gaming contexts, in this paper we currently consider only visibility, colors or textures will be incorporated in the future for a more comprehensive viewpoint evaluation.

In relation to the evaluation of the gain of information in the selection of the point of view, Delmerico et al. [23] propose a comparison of volumetric information metrics for the active reconstruction of 3D objects. Its approach is based on the emission of rays in the 3D space of voxels and the entropy calculation to evaluate the information gain of a particular view.

Zhang and Fei [24] classify viewpoint selection methods into three main categories. The methods based on geometrical information such as the one by Vazquez et al. [14] consider measures such as the area, the projected area, the silhouette and other characteristics of the viewpoint, but they can overlook the structural information of the 3D model. Methods based on visual characteristics focus on visual attributes such as silhouette, curvature [25], and mesh importance [26]. Although these methods are efficient for measuring visual characteristics, they can omit important geometric information in the scene. Finally, the methods based on semantics as proposed in [27,28] evaluate the point of view through the use of semantic segmentation, which considers semantic components of the scene and artificial labels. However, automatic segmentation can be challenging and require manual intervention.

Kullback–Leibler divergence has been used in visualization and computer graphics areas before. Bordoloi et al. [17] have introduced a method for enhancing the effectiveness of volume rendering by guiding users towards informative viewpoints obtained with viewpoint entropy. Kullback–Leibler divergence is employed as a measure of dissimilarity between probability distributions associated with viewpoints. Ruiz et al. [29] introduced a framework for obtaining transfer functions for the volumetric data based on user-provided target distributions. The transfer functions are derived by minimizing the Kullback–Leibler distance between visibility distribution from viewpoints and user-selected target distributions. Lan et al. [30] built a robotic photography system to find the optimal viewpoint of a scene. The system assesses aesthetic composition by comparing, with Kullback–Leibler divergence, the distribution of a current composition with a model or target composition. Smaller Kullback–Leibler divergence values indicate a more aesthetically pleasing composition. Furthermore, Yokomatsu et al. [31] introduce an autonomous indoor drone photographer that searches for a viewpoint in 3D space employing a Gaussian mixture model to represent subjects on its camera screen. Using variational Bayes clustering for four or more subjects, it evaluates the composition through Kullback–Leibler divergence against a user-defined reference based on user-set composition rules. To the best of our knowledge, although K-L divergence has been used for various applications, it has not been used for viewpoint selection in open environments and multi-object scenes, neither total variation nor χ^2-divergence have been used in viewpoint selection, in general.

In the field of video games, the selection of the best viewpoint for virtual scenes has received less attention compared to robotics. There are few studies that specifically address this topic in video games, especially those that work with virtual scenes in real time. An example of this is the work conducted by Galvane [32], where he proposes a system based on the Reynolds steering behavior model to control and coordinate a collection of autonomous camera agents that move in dynamic 3D environments with the objective of filming events of multiple scales. In addition, Galvane proposes an approach based on the importance of cinematographic reproduction in games, taking advantage of the narrative and geometric information to automatically calculate the trajectories and the planning of the camera in interactive time. The best camera viewpoints are selected based on a function that takes into account symbolic projection, narrative importance, narrative relevance, and visual quality. Virtual camera rails are then created to guide camera movements throughout the scene, and camera movements are calculated, optimizing the trajectory to achieve smooth transitions.

Another approach proposed by Lino and Christie [33] is the use of a theoretical surface model that efficiently generates a variety of viewpoints corresponding to the exact on-screen composition of two or three objectives. These approaches, based on algebraic models, offer fast and efficient solutions for the automatic calculation of points of view, although they tend to be limited to a small number of objectives and may not address cinematographic problems such as obstruction or occlusion.

The use of neural networks has also been proposed for the selection of viewpoints in 3D environments. Zhang et al. [34] presented an optimization strategy for computing high-quality virtual viewpoints for aesthetic images by combining a multi-branch CNN and a viewpoint correction method, integrating visual perception with the calculation of geometric information. Furthermore, deep learning has been utilized to reconstruct the 3D pose from the image obtained in the video by Kiciroglu et al. [35]. An algorithm was created that, based on the camera position, calculates the uncertainty and generates a set of future camera positions, taking into account that the scene is unknown. The authors have then used neural networks to return the 3D human pose from monocular images.

Hartwig et al. [36] introduced a neural view quality measure aligned with human preferences. The study demonstrated that this measure generalized not only to models unseen during training but also to unseen model categories.

2.2. f-Divergences

f-divergences generalize the Kullback–Leibler divergence between two distributions [37–39]. Given p, q distributions $\{p_i\}_{i=1}^n, \{q_i\}_{i=1}^n, \forall i : p_i, q_i \geq 0$, f-divergence $D_f(p,q)$ is defined as:

$$D_f(p,q) = \sum_{i=1}^{n} q_i f(p_i/q_i),$$

where $f(x)$ is a real-valued convex function such that $f(1) = 0$. By definition $0f(0/0) = 0$, and for $a > 0, 0f(a/0) = \lim_{t^+ \to 0} tf(\frac{a}{t})$.
$D_f(p,q)$ holds the properties:

- $D_f(p,q) \geq 0$, and $p \equiv q \implies D_f(p,q) = 0$ ($D_f(p,q) = 0 \implies p \equiv q$ for $f(x)$ strictly convex).
- $D_f(p,q)$ is convex in both p and q.
- $D_f(p,q) \equiv D_{f(x)+c(x-1)}(p,q)$.
- Given a transform T, $D_f(p,q) \geq D_f(pT, qT) = 0$ (data processing inequality-DPI). In particular, T can be any clustering of indexes.

Well known examples of f-divergences are K-L divergence or relative entropy, with $f(x) = x \log x$, $K\text{-}L(p,q) = \sum_{i=1}^n p_i \log \frac{p_i}{q_i}$, total variation, with $f(x) = 1/2|x-1|$, $TV(p,q) = \frac{1}{2}\sum_{i=1}^n |p_i - q_i|$, and which is the only f-divergence that is also a Euclidian distance, squared-Hellinger distance, with $f(x) = (\sqrt{x}-1)^2$, $H^2(p,q) = \sum_{i=1}^n(\sqrt{p_i} - \sqrt{q_i})^2$ and χ^2-divergence, with $f(x) = (x-1)^2$, $\chi^2(p,q) = \sum_{i=1}^n \frac{(p_i - q_i)^2}{q_i}$. Another example is the one-parameter family of f-divergences, the Tsallis divergences, with $\alpha > 0$,

$$T_\alpha(p,q) = \frac{1}{1-\alpha}\left(\sum_{i=1}^n \frac{p_i^\alpha}{q_i^{\alpha-1}} - 1\right),$$

which includes for $\alpha = 2$ the χ^2 divergence, for $\alpha = 1/2$ the squared Hellinger distance, and for $\alpha = 1$ by continuity the K-L divergence [40].

Observe that Shannon entropy, $H(p) = -\sum_{i=1}^n p_i \log p_i$ is related to the Kullback–Leibler divergence when q distribution is uniform, i.e., when $q = \{\frac{1}{n}\}_{i=1}^n$,

$$K\text{-}L(p, \{1/n\}) = \sum_{i=1}^n p_i \log \frac{p_i}{1/n} = \log n \sum_{i=1}^n p_i - \sum_{i=1}^n p_i \log p_i = \log n - H(p).$$

3. Proposed Method

The intuition behind our method is to measure the visibility of the objects from the camera and compare the distribution of visibility to the distribution of the areas using f-divergences. In this aspect, the more informative view would be when each object is visible proportionally to its area. The lower the divergence value, the more proportional are visibility and relative area distributions.

3.1. Visibility

In the field of computer graphics and video game development, two fundamental concepts related to visual perception and projection are the Field of View (FOV) and the Frustum. The Field of View refers to the angular range of the observable scene or the visual range of a camera in a three-dimensional environment. A wider FOV provides a more extensive peripheral vision, while a narrower FOV focuses on a smaller area with greater detail. On the other hand, the Frustum represents the shape of a truncated pyramid.

In previous work by Rigau et al. [41], the visibility of a point in a 3D scene was studied based on information theoretic criteria. This work proposed to use the Kullback–Leibler divergence between the solid angles projected by the objects and the unoccluded projections as a measure of the viewpoint. Also, Sbert et al. [5] proposed as a viewpoint measure of an object the K-L divergence between the projected areas of the triangles of the object mesh and the true areas. We extend these ideas to consider the visibility measured by the form factors from the viewpoint [7,8], constrained to the camera frustum, and we consider two additional divergence measures in addition to K-L divergence.

Given a scene $S \subset R^3$, and O the set of objects $\{o_i\}$ in the scene. We define A_i as the area of object o_i, A_T as the total area of the objects in the scene plus background area, and $a_i = \frac{A_i}{A_T}$ as the relative area of object o_i. Given the position $x \in S$ of the camera, we consider dA_x in the normal plane to the camera direction. Given a point y in the surface of an object in O, dA_y is in the tangent plane at y, θ_y is the angle between the normal to dA_y with the line that joins x and y, θ_x is the angle with the normal at dA_x, and $d(x,y)$ the distance between x and y (see Figure 1).

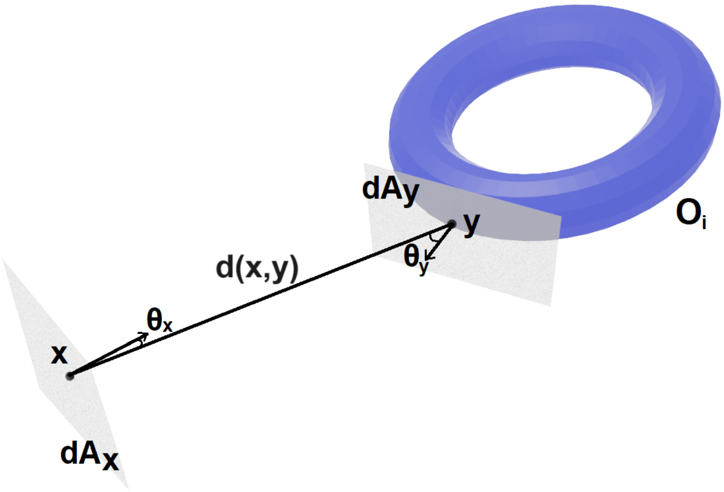


Figure 1. Notation used for differential form-factor, x is the camera position, dA_x is on the camera plane, y is a point on the surface of object o_i at distance $d(x,y)$ from x, dA_y is on the tangent plane at y, θ_x and θ_y are the angles between the normals at x and y and the line joining x and y respectively.

3.2. Hemisphere Form-Factors

$F(dA_x, dA_y)$ is the differential form factor (or measure of the visibility) between differential areas dA_x, dA_y at point x. It forms a continuous probability distribution, i.e.,

$$\int_{y \in O} F(dA_x, dA_y) dA_y = 1,$$

where:

$$F(dA_x, dA_y) dA_y = \frac{\cos\theta_x \cos\theta_y}{\pi d(x,y)^2} v(x, dA_y) dA_y, \tag{1}$$

and $v(x, dA_y)$ is a binary visibility function (equal to 1 if x and y are mutually visible and 0 otherwise). If $d\omega$ is the area subtended by dA_y,

$$F(dA_x, dA_y) d\omega = \frac{1}{\pi} v(x, dA_y) \cos\theta_x d\omega, \tag{2}$$

taking into account that $d\omega = \frac{\cos\theta_x \cos\theta_y}{d(x,y)^2} dA_y$, and the π factor is the normalization constant as the integral over the hemisphere is $\int_{\Omega/2} \cos\theta_x d\omega = \pi$. The form factor for object i around the hemisphere centered in dA_x is then defined as

$$F(dA_x, o_i) = \int_{\Omega/2} \frac{1}{\pi} v(x, o_i(\omega)) \cos\theta_x d\omega = \int_{\Omega_i} \frac{1}{\pi} \cos\theta_x d\omega, \tag{3}$$

where $v(x, o_i(\omega))$ is 1 or 0 depending on whether object o_i is visible or not from direction ω, and Ω_i is the solid angle over the hemisphere around dA_x from which the object i is visible. If $\Omega_b = \Omega/2 - \sum_i \Omega_i$ is the solid angle projected by the background, we can write as only one object (or the background) is visible in a given direction ω

$$\begin{aligned} \int_{\Omega/2} \frac{1}{\pi} \cos\theta_x d\omega = 1 = \sum_i \int_{\Omega_i} \frac{1}{\pi} \cos\theta_x d\omega + \int_{\Omega_b} \frac{1}{\pi} \cos\theta_x d\omega \\ = \sum_i F(dA_x, o_i) + F(dA_x, o_b), \end{aligned} \tag{4}$$

where $F(dA_x, o_b)$ is the background form-factor. However, the whole hemisphere is not visible through a virtual camera, thus in the next section, we restrict the visibility to the camera frustum.

3.3. Frustum Form-Factors and f-Divergence Frustum Viewpoint Measures

Let us consider the visibility restricted to a frustum fr that subtends a solid angle Ω_{fr} around dA_x. The normalization constant k_{fr} for fr form-factors would be

$$k_{fr} = \int_{\Omega_{fr}} \cos\theta_x d\omega$$

and can be computed by importance sampling Monte Carlo integration [42]. We can use for instance N rays distributed around the hemisphere with probability density function (pdf) $p(\omega) = 1/\pi \cos\theta_x d\omega$, in which case $k_{fr} \approx \pi N_{fr}/N$, where N_{fr} is the number of rays crossing the frustum. We define the frustum form factor as

$$F(dA_x, o_i) = \int_{\Omega_i \cap \Omega_{fr}} 1/k_{fr} \cos\theta_x d\omega, \tag{5}$$

Equation (5) can be computed efficiently by Monte Carlo by casting rays distributed according to $p(\omega) = 1/\pi \cos\theta_{x_i}$ and simply counting the fraction of rays within the frustum and hitting the object o_i (see Appendix A). This can be conducted for all objects

$\{o_i\}$ at the same time. Let a background object be o_b, for instance, a background hemisphere, then we have that

$$\sum_{i=1}^{n} F(dA_x, o_i) + F(dA_x, o_b) = 1.$$

Let us now consider the fraction of the total area corresponding to each object, say a_i. For the background hemisphere, we have a_b. Indeed

$$\sum_{i=1}^{n} a_i + a_b = 1.$$

Then we can consider the f-divergence measures between the form-factor distribution and the relative area distribution.

3.3.1. Kullback–Leibler Divergence

We will consider first the Kullback–Leibler divergence between the two distributions,

$$K\text{-}L(dA_x, \{o_i\}) = \sum_{i=1}^{n} F(dA_x, o_i) \log \frac{F(dA_x, o_i)}{a_i} + F(dA_x, o_b) \log \frac{F(dA_x, o_b)}{a_b}, \tag{6}$$

where if, for some i, $F(dA_x, o_i) = 0$, we take continuity $F(dA_x, o_i) \log \frac{F(dA_x, o_i)}{a_i} = 0$. And using the hit count, where N_i is the number of hits on object i, and N_b on the background, the frustum form factor is approximated by

$$F(dA_x, o_i) \approx \frac{N_i}{N_{fr}}, \tag{7}$$

and thus the Kullback–Leibler divergence is approximated by

$$K\text{-}L(dA_x, \{o_i\}) \approx \sum_{i=1}^{n} \frac{N_i}{N_{fr}} \log \frac{\frac{N_i}{N_{fr}}}{a_i} + \frac{N_b}{N_{fr}} \log \frac{\frac{N_b}{N_{fr}}}{a_b}. \tag{8}$$

Observe that $a_i = A_i/(A_T + A_b)$, and $a_b = A_b/(A_T + A_b)$, and $A_T = \sum_i A_i$.
Now, taking the Kullback–Leibler divergence as a viewpoint measure has one problem: it does not penalize not-seen objects, on the contrary, the corresponding term in the sum is 0. Let us then consider two alternatives.

3.3.2. Total Variation and χ^2-Divergence Frustum Viewpoint Measures

Now, using the total deviation as a frustum viewpoint measure, we obtain

$$TV(dA_x, \{o_i\}) \approx 1/2\Big(\sum_{i=1}^{n} |\frac{N_i}{N_{fr}} - a_i| + |\frac{N_b}{N_{fr}} - a_b|\Big). \tag{9}$$

And using χ^2-divergence,

$$\chi^2(dA_x, \{o_i\}) \approx \sum_{i=1}^{n} \frac{(\frac{N_i}{N_{fr}} - a_i)^2}{a_i} + \frac{(\frac{N_b}{N_{fr}} - a_b)^2}{a_b}. \tag{10}$$

Observe that in both measures an object o_i non-visible adds the same amount a_i. This is the main difference with respect to the K-L measure, where the race of a non-visible object disappears.

Another approach, proposed by Lino and Christie [33], is the use of a theoretical surface model that efficiently generates a variety of viewpoints corresponding to the exact on-screen composition of two or three objectives. These approaches based on algebraic models offer fast and efficient solutions for the automatic calculation of points of view, although they tend to be limited to a small number of objectives and may not address cinematographic problems such as obstruction or occlusion.

3.4. Background Issues and Importance

Suppose we are in an open scene with no background surface(s) to consider. How do we deal with this case? On the one hand, we consider the rays missing the objects $\{o_i\}$ as hitting the background, and we count them as N_b. Now, instead of considering some fictitious background surface as a hemisphere enveloping the objects we can decide a priori how much background we want to see in our frustum, and simply set a_b as a proportion of $\sum_{i=1}^{n} a_i$. A small $\frac{a_b}{\sum_i a_i}$ value means that our viewpoint measure will favor a small background proportion of the frustum, while a big one will mean the reverse. This can be extended to any object in $\{o_i\}$. It can be formalized, in a similar way to [21], by defining importance non zero values $\{p_i\}$ for objects and background p_b and considering the new pseudo-area distribution $a'_i = \frac{a_i p_i}{\sum_k a_k p_k + a_b p_b}$, $a'_b = \frac{a_b p_b}{\sum_k a_k p_k + a_b p_b}$.

3.5. Total Surface vs. Visible Surface

We have considered in the previous sections $\{a_i\}$ as the relative surface area of objects $\{o_i\}$. But for each object, we could have considered the visible area $\{a'_i\}$, where evidently $\forall i, a'_i \leq a_i$. This would make sense if objects have an important share of hidden parts. The $\{a'_i\}$ could be computed in a preprocess, using for instance global uniformly distributed lines [8]. A similar discussion between visible and total areas can be found in [21].

3.6. Particular Cases with Kullback–Leibler Divergence

Let us suppose that only a single object o_i is visible through a particular frustum. Then, $N_i/N_{fr} = 1$, and the K-L divergence would be $-\log a_i$. Observe that this value is independent of how near the object is, as far as it covers the whole frustum. For instance, suppose $a_i = 1/32$, then the K-L divergence would be equal to 5. The same would happen if only the background is visible, the K-L value would be $-\log a_b$. For instance, giving low importance to the background, say $a_b = 1/64$, the K-L value would be equal to 6, while giving it much higher importance, say $a_b = 1/2$, the K-L value would be equal to 1.

3.7. K-L-Divergence Frustum Viewpoint Measure versus Frustum Viewpoint Entropy

Analog to the classic viewpoint entropy [14], we define frustum viewpoint entropy as

$$VE(dA_x, \{o_i\}) = -(\sum_{i=1}^{n} F(dA_x, o_i) \log F(dA_x, o_i) + F(dA_x, o_b) \log F(dA_x, o_b)), \quad (11)$$

with n as the number of objects $\{o_i\}$. The best view according to this measure would be the one with the highest value, which corresponds to all objects and backgrounds seen equally, all form factors are equal, independent of their relative areas.

Observe that if we take the K-L divergence frustum viewpoint for all objects including background having the same relative area, or identically if we take the importance of each object including a background as inversely proportional to their area, the relative area for all objects and background is now $\frac{1}{n+1}$, thus we have

$$\begin{aligned} K\text{-}L(dA_x, \{o_i\}) &= \sum_{i=1}^{n} F(dA_x, o_i) \log \frac{F(dA_x, o_i)}{\frac{1}{n+1}} + F(dA_x, o_b) \log \frac{F(dA_x, o_b)}{\frac{1}{n+1}} \\ &= \sum_{i=1}^{n} F(dA_x, o_i) \log F(dA_x, o_i) + F(dA_x, o_b) \log F(dA_x, o_b) \\ &+ \log(n+1)(\sum_{i=1}^{n} F(dA_x, o_i) + F(dA_x, o_b)) \\ &= -VE(dA_x, \{o_i\}) + \log(n+1). \end{aligned} \quad (12)$$

This is, in that particular case one can indistinctly use K-L divergence or entropy measure, just now, the best views with K-L measure would be the ones with the lowest value, and for entropy the ones with the highest value.

3.8. TV Changes Smoothly

The fact that TV is a Euclidean distance allows us to bound the increment of TV measure when we consider another frustum or when we change the relative areas (for instance background area). Suppose we change from a frustum with form-factors $\{F_i\}$ to another frustum with form-factors $\{F_i'\}$ (or the same frustum but objects have moved position). Suppose also that we change areas from $\{a_i\}$ to $\{a_i'\}$ (for instance, changing background area). Then, we can bound the change in TV measure. Effectively, using the symmetric and triangular inequality properties of a Euclidean distance we can state the following inequalities,

$$\begin{aligned} TV(F,a') &\geq TV(F,a) - TV(a,a'), \\ TV(F,a') &\leq TV(F,a) + TV(a,a'), \\ TV(F',a) &\geq TV(F,a) - TV(F,F'), \\ TV(F',a) &\leq TV(F,a) + TV(F,F'). \end{aligned}$$

Observe that these bounds imply that a small change in the form factors or in the area implies a small change in the TV viewpoint measure, i.e., it changes smoothly. This is not always the case with K-L and χ^2 measures.

3.9. Rays vs. Projection

Projection has been used in the past to compute viewpoint measures of a 3D object or to simplify its mesh based on a viewpoint measure. Could we equally use in the context of this paper projection instead of casting rays to compute the f-divergence frustum viewpoint measure? Let us remember first that we base our viewpoint measure on the form factor measuring the visibility. Before switching to ray casting, form factors were computed via projecting on the five faces of a hemicube [43], where the pixels of the faces had unequal weights. This gave an approximation to the actual value of the form factor. Implementing hemicube in a game engine would be tricky. Sillion and Puech [44] computed form factors by substituting the five projections with a single projection, but objects near the horizon were missed. On the other hand, ray casting can compute the actual values, up to the statistical error, and also each ray can be computed independently, as the Monte Carlo method is by nature parallelizable. In addition, it is very simple to program and game engines support it in real time.

3.10. Implementation

The proposed method has been implemented in the Unity game engine making use of its ray-casting routines.

4. Evaluation

To check the correct computation of the form factor, we devised a configuration where the form factor can be computed analytically, see Figure 2. The results are shown in Table 1. As a form-factor F_i computed with random rays corresponds to a binary hit or miss distribution, its variance or expected quadratic error is $\frac{F_i(1-F_i)}{N_{rays}}$, and thus the expected error is $\sqrt{\frac{F_i(1-F_i)}{N_{rays}}}$, which is compared in the table with the experimental error.

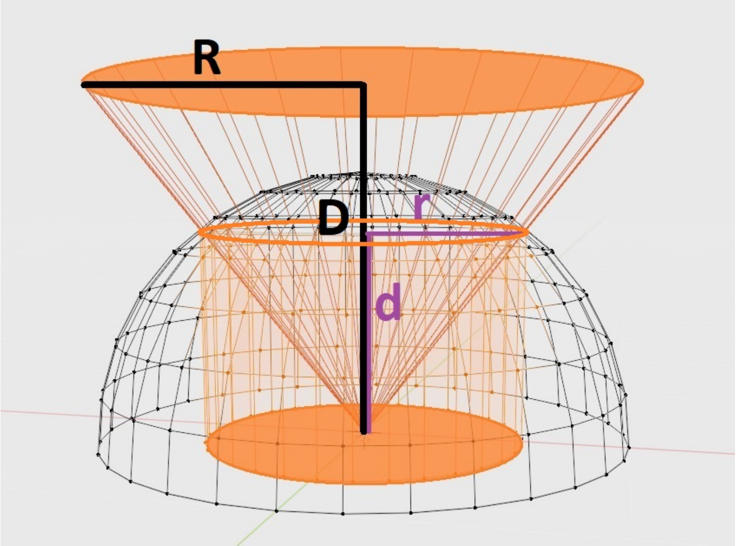


Figure 2. To check the correct distribution of cast lines, we compare the experimental result with the analytically computed form factor corresponding to a disk (in orange color) with radius R at distance D from the camera and orthogonal to its plane. The form factor corresponds to the area of the projected circle, πr^2, divided by the area of the disk with radius 1, π, this is $\frac{\pi r^2}{\pi} = r^2$. By similarity of the triangles, $k = \frac{d}{r} = \frac{D}{R}$. As $d^2 + r^2 = 1$, we can easily find that $r^2 = \frac{1}{k^2+1}$, and thus the form factor is $\frac{1}{k^2+1}$.

Table 1. Form factor computation validation, see Figure 2. The frustum covers the whole hemisphere. The expected error $\sqrt{F_i(1-F_i)/N_{rays}}$ when form-factor = 0.2 and for 10,000 rays is 0.004, for 100,000 rays is 0.00126. For form-factor = 0.1, for 10,000 rays is 0.003, for 100,000 rays is 0.000948.

Values: $\boldsymbol{R = 0.5}$, $\boldsymbol{D = 2}$, $\boldsymbol{k = 4}$. Analytical Form-Factor = **0.2**				
Total Rays	**Hit Disc Rays**	**Form-Factor**	**Abs. Error**	**Expected Error**
10,000	2017	0.2017	0.0017	0.004
10,000	2024	0.2024	0.0024	0.004
10,000	2008	0.2008	0.0008	0.004
100,000	19,771	0.19771	0.00229	0.00126
100,000	19,825	0.19825	0.00175	0.00126
100,000	19,835	0.19835	0.00165	0.00126
Values: $\boldsymbol{R = 1}$, $\boldsymbol{D = 3}$, $\boldsymbol{k = 3}$. Analytical form-factor = **0.1**				
Total rays	Hit disc rays	Form-factor	Abs. Error	Expected error
10,000	975	0.0975	0.0025	0.003
10,000	982	0.0982	0.0018	0.003
10,000	1020	0.1020	0.0020	0.003
100,000	9736	0.09736	0.00264	0.000948
100,000	9851	0.09851	0.00149	0.000948
100,000	9925	0.09925	0.00075	0.000948

To study the error in the three viewpoint measures considered we compute them for Figure 3 with varying number of rays. As we can see in Table 2, by casting 100,000 rays the change in the results between the different iterations is less than 0.01, thus a difference of 0.02 will be considered significant to compare two viewpoints.

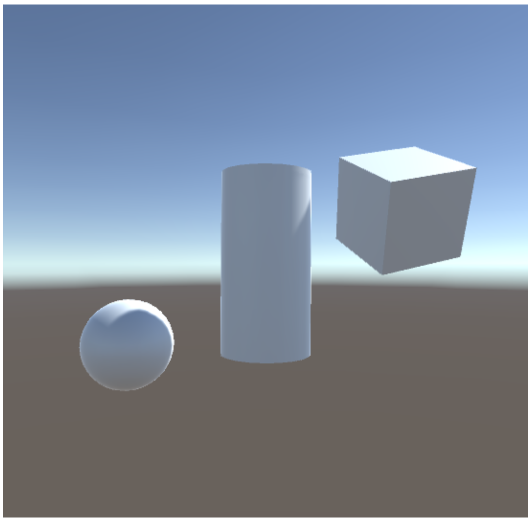
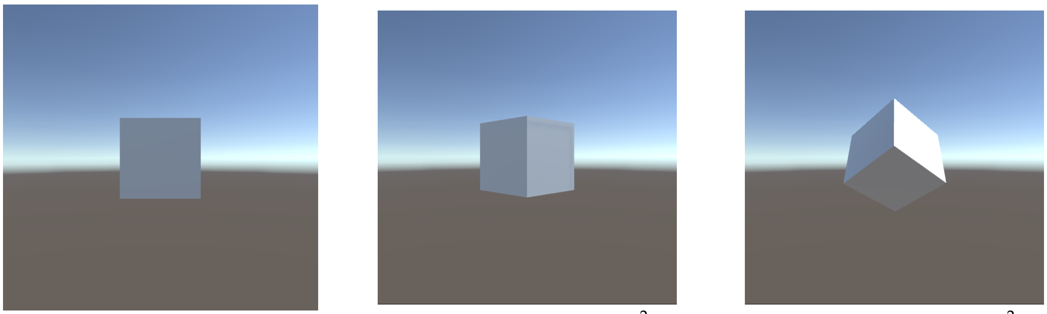


Figure 3. Scene used for the measurements in Table 2.

Table 2. Values of the measures for Figure 3 for different number of rays.

Total Rays	K-L	TV	χ^2
10,000	0.3466502	0.331814	0.4405105
10,000	0.3530622	0.3346139	0.4405105
10,000	0.2731511	0.3346139	0.3538287
10,000	0.2862652	0.304014	0.369765
10,000	0.3157122	0.3179139	0.4044647
100,000	0.3179567	0.3190739	0.4072793
100,000	0.3164447	0.318334	0.4054418
100,000	0.3170037	0.318584	0.4060861
100,000	0.321581	0.320724	0.4115101
100,000	0.3169065	0.318514	0.4059337

Then, given that the background area was added as a variable due to the fact that the background has no area in Unity, different tests were carried out to evaluate if there is a suitable background area percentage in general. The first scene considered was a single cube, rotated to obtain three different views of it, see Figure 4, and considering the background area values of 1%, 25%, 50%, 75% and 99% of the total area.



K-L	TV	χ^2
1%: 5.560	0.671	80.9
25%: 1.426	0.496	2.33
50%: 0.537	0.3015	0.646
75%: 0.12	0.1195	0.135
99%: 0.185	0.0625	0.699

K-L	TV	χ^2
1%: 5.395	0.6585	77.857
25%: 1.332	0.481	2.194
50%: 0.488	0.2895	0.596
75%: 0.09	0.1055	0.105
99%: 0.244	0.075	1.008

K-L	TV	χ^2
1%: 5.377	0.657	77.54
25%: 1.289	0.474	2.13
50%: 0.472	0.2855	0.579
75%: 0.078	0.099	0.093
99%: 0.299	0.086	1.331

Figure 4. Comparison of measures (TV: Total Variation, K-L: Kullback–Leibler, and χ^2) by rotating a single cube, and computed with 100,000 rays. Percentages are the relative area assigned to background.

We note that if we assign to the background a relative area equal to its experimental form factor for each view, the measures would give the same result while seeing the objects from far or near. Thus, to avoid this, the relative area of the background should be kept fixed at a relative percentage of the total area.

We observe that for background areas of 1%, 25%, 50%, and 75%, the value of the three measures decreases as the number of visible faces of the cube increases. The only case where the measure increases when the number of visible faces of the cube increases corresponds to the background 99% of the total area, which makes sense considering that increasing the number of visible faces decreases the area occupied by the background.

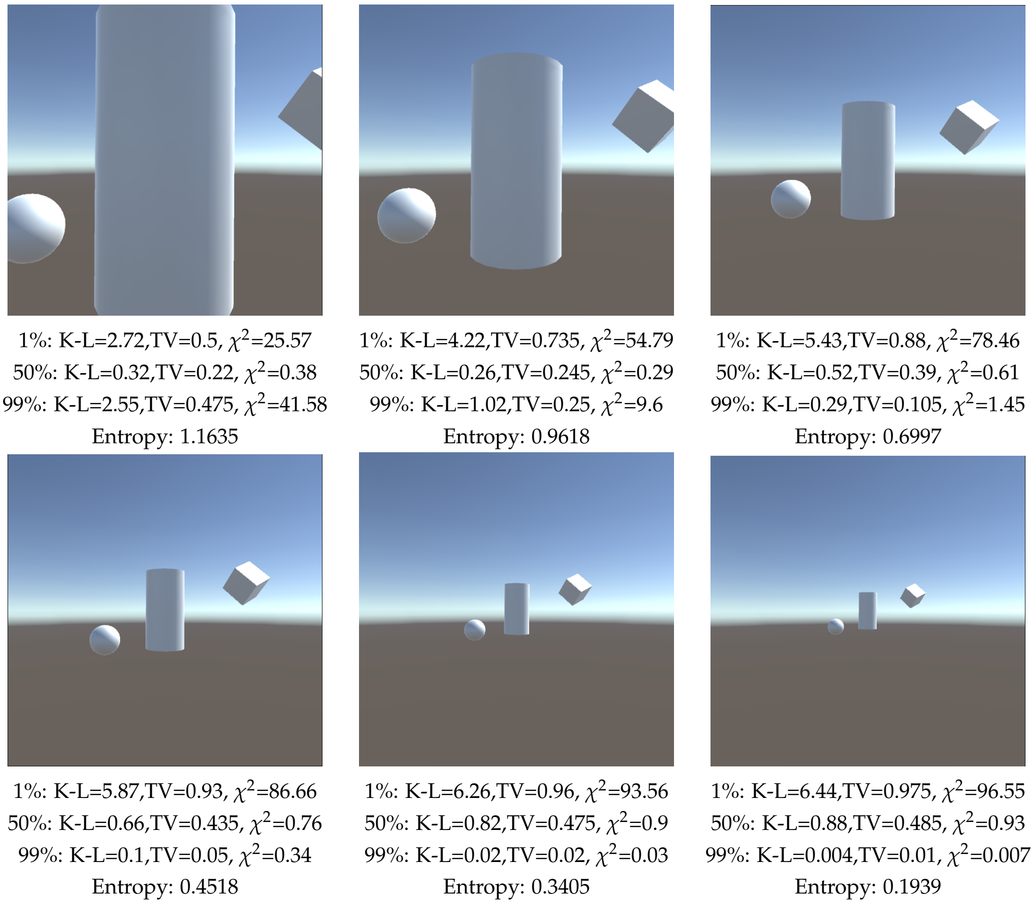
Then, a scene composed of a cube, a cylinder and a sphere, Figure 5, is analyzed, placed in the center of the scene and zoomed in and out, and for values of the background area of 1%, 50%, and 99% of the total area. We also compute viewpoint entropy. When analyzing the values assigned to the background, we notice that by assigning to it a low value (1%), the lowest measure values will be when the more percentage of the FOV of the camera is occupied by objects, while zooming in the lowest value will be the ones in Figure 5 upper left image for all the measures. Analyzing the case where the background value is 99%, the measurements will decrease by zooming out the camera, thus, the measurement is minimal in Figure 5 lower right. Concerning the viewpoint entropy measure, remember that its behavior is inverse to the one of the K-L measures; the higher value would correspond to a better view. It decreases when zooming out, and increases with zooming in, which is correct. However, in Figure 5 upper left, when we zoom in on the cylinder object, entropy is still growing, which is not the expected behavior for a good view, as it is clearly a worse view than Figure 5 upper center. On the other hand, by assigning to the background an area of 50% of the total area, K-L and χ^2 measures achieve a minimum in Figure 5 upper center, which we consider to be the best view. The TV measure decreases too when zooming in, but its minimum is not achieved in Figure 5 upper center but in Figure 5 upper left.

As a third example, as seen in Figure 6, the camera was rotated around a vertical axis obtaining different views of a scene composed of a cube, a cylinder and a sphere. For a 1% background value as well as for 50%, the lowest value for the three divergences happened when the three objects were visible, while for a background value of 99% the lowest value for the divergences happened when no object was visible. For Shannon entropy, the highest value is found when all objects are visible while the lowest value is when no object is visible. Observe that the behavior of the entropy measure is inverse to the one of the K-L divergence. Remember that entropy does not take into account the area of objects and background, and behaves inversely as the K-L divergence when all objects, including background, have the same relative area, thus background value does not play any role in the entropy measure computation. In this example, the entropy measure gives results coherent with our expectation of a good view.

From our results, we infer that assigning the background importance of 50% gives good results.

To test the measures in the presence of occlusion, we employed a scene composed of a cylinder, a sphere, and a cube. The scene was observed from lateral (Figure 7 left), diagonal (Figure 7 center), and frontal (Figure 7 right) views while maintaining a constant distance to the cylinder (Figure 7). The results show that K-L identifies as the best viewpoint the diagonal view, where a larger area is visible and there is minimal occlusion between objects. Additionally, it designates the lateral view as the least favorable, where greater occlusion among objects is present. Similarly, χ^2 and Shannon entropy perform well; TV, while identifying the diagonal view as the best, does not distinguish between the lateral and frontal ones.

Finally, in Figure 8 we find two scenes with five objects each, the scenes only differ in the rotation of one cube. While K-L divergence clearly appreciates an improvement in the second view (Figure 8 right), TV (see Section 3.8) and χ^2 divergences and Shannon Entropy improvement falls within the threshold we established of 0.02.



1%: K-L=2.72,TV=0.5, χ^2=25.57
50%: K-L=0.32,TV=0.22, χ^2=0.38
99%: K-L=2.55,TV=0.475, χ^2=41.58
Entropy: 1.1635

1%: K-L=4.22,TV=0.735, χ^2=54.79
50%: K-L=0.26,TV=0.245, χ^2=0.29
99%: K-L=1.02,TV=0.25, χ^2=9.6
Entropy: 0.9618

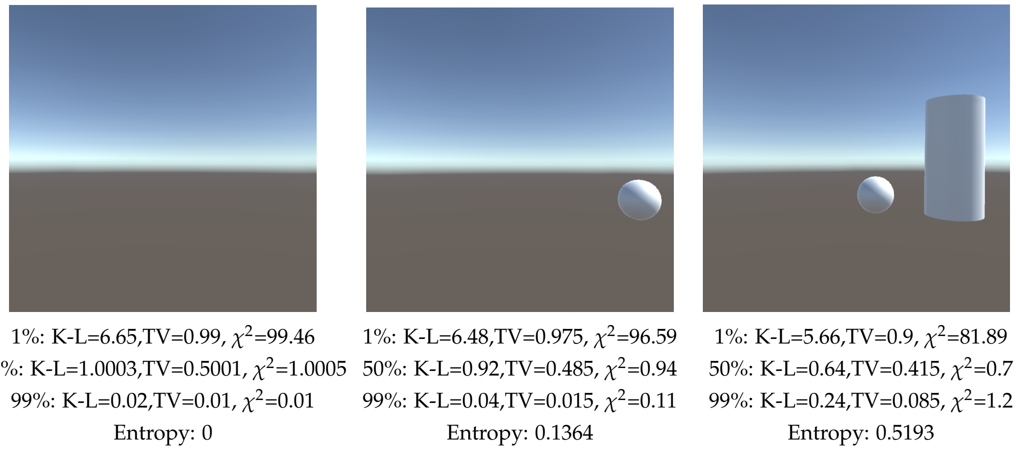
1%: K-L=5.43,TV=0.88, χ^2=78.46
50%: K-L=0.52,TV=0.39, χ^2=0.61
99%: K-L=0.29,TV=0.105, χ^2=1.45
Entropy: 0.6997

1%: K-L=5.87,TV=0.93, χ^2=86.66
50%: K-L=0.66,TV=0.435, χ^2=0.76
99%: K-L=0.1,TV=0.05, χ^2=0.34
Entropy: 0.4518

1%: K-L=6.26,TV=0.96, χ^2=93.56
50%: K-L=0.82,TV=0.475, χ^2=0.9
99%: K-L=0.02,TV=0.02, χ^2=0.03
Entropy: 0.3405

1%: K-L=6.44,TV=0.975, χ^2=96.55
50%: K-L=0.88,TV=0.485, χ^2=0.93
99%: K-L=0.004,TV=0.01, χ^2=0.007
Entropy: 0.1939

Figure 5. Comparison of measures (TV: Total Variation, K-L: Kullback–Leibler, χ^2 and Shannon entropy) when zooming out the camera, and computed with 100,000 rays. Percentages are the relative area assigned to background.

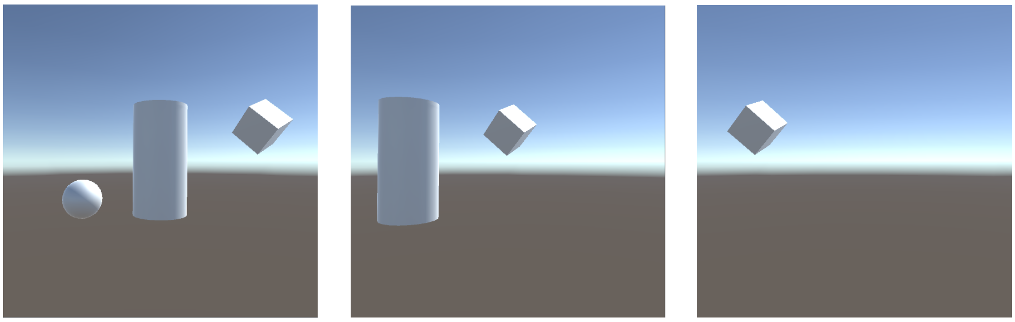


1%: K-L=6.65,TV=0.99, χ^2=99.46
50%: K-L=1.0003,TV=0.5001, χ^2=1.0005
99%: K-L=0.02,TV=0.01, χ^2=0.01
Entropy: 0

1%: K-L=6.48,TV=0.975, χ^2=96.59
50%: K-L=0.92,TV=0.485, χ^2=0.94
99%: K-L=0.04,TV=0.015, χ^2=0.11
Entropy: 0.1364

1%: K-L=5.66,TV=0.9, χ^2=81.89
50%: K-L=0.64,TV=0.415, χ^2=0.7
99%: K-L=0.24,TV=0.085, χ^2=1.2
Entropy: 0.5193

Figure 6. *Cont.*

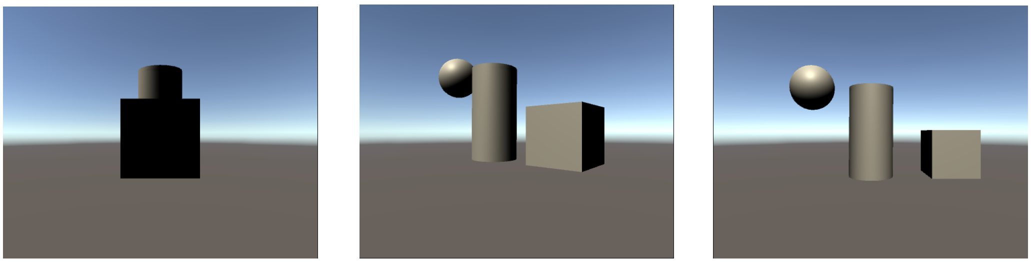


1%: K-L=5.45,TV=0.38, χ^2=78.82
50%: K-L=0.52,TV=0.395, χ^2=0.62
99%: K-L=0.26,TV=0.1, χ^2=1.25
Entropy: 0.697

1%: K-L=5.64,TV=0.9, χ^2=82.02
50%: K-L=0.59,TV=0.405, χ^2=0.67
99%: K-L=0.22,TV=0.085, χ^2=1.03
Entropy: 0.5759

1%: K-L=6.41,TV=0.97, χ^2=95.62
50%: K-L=0.9,TV=0.485, χ^2=0.93
99%: K-L=0.04,TV=0.015, χ^2=0.08
Entropy: 0.1786

Figure 6. Comparison of measures (TV: Total Variation, K-L: Kullback–Leibler, χ^2 and Shannon entropy) when rotating the camera around a vertical axis, and computed with 100,000 rays. Percentages are the relative area assigned to background.

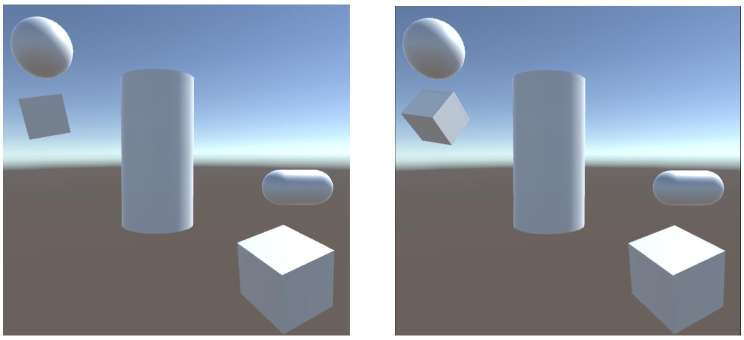


K-L=0.5285,TV=0.3603, χ^2 =0.5624
Entropy=0.6754

K-L=0.3525 ,TV=0.3289, χ^2=0.4388
Entropy=0.895

K-L=0.4358 ,TV=0.3672, χ^2=0.5397
Entropy=0.7617

Figure 7. Comparison of measures (TV: Total Variation, K-L: Kullback–Leibler, and χ^2) for occlusion, and computed with 100,000 rays. Background area = 50%.



50%: K-L=0.2969,TV=0.2742, χ^2=0.3413
Entropy= 0.8607

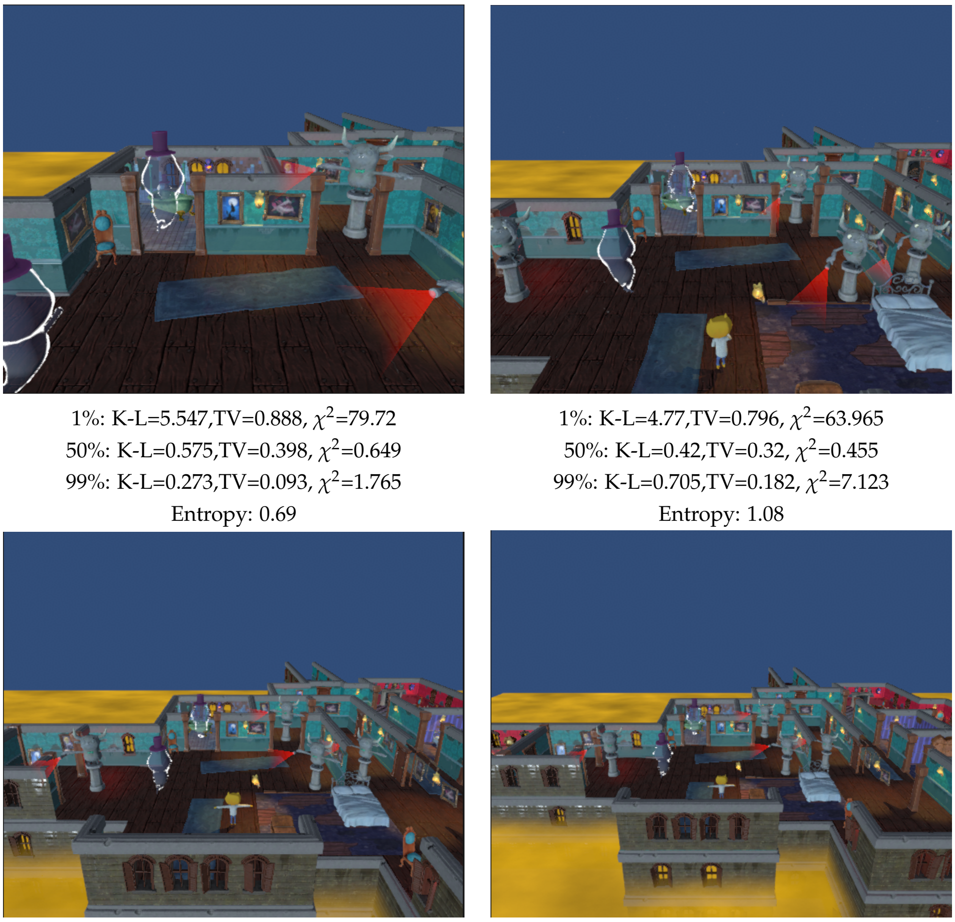
50%: K-L= 0.2744,TV=0.2681, χ^2=0.323
Entropy= 0.8702

Figure 8. Comparison of measures (Entropy, TV: Total Variation, K-L: Kullback–Leibler and χ^2) when rotating one single object, and computed with 100,000 rays. Percentages are the relative area assigned to background.

4.1. Validation in a Video Game Environment

To validate the implementation in a video game, we tested the method in the John Lemmon Unity game [45]. The first two evaluated scenes, depicted in Figures 9 and 10, consist of the main character (a yellow-headed kitty), six enemies (three grey gargoyles holding a red lantern in their hand and three grey ghosts with a purple hat), and the background. Similar to the analysis in Figure 5, we initially examined what happens when the camera is zoomed in and out of the scene for background area values of 1%, 50%, and

99% of the total area. As before, assigning a low value (1%) to the background area results in lower measurement values when a higher percentage of the camera's field of view is occupied by objects. In this case, it corresponds to the upper-right image in Figure 9, while the highest value occurs when the background occupies the largest percentage, as seen in the lower-right image in Figure 9. Analyzing the case where the background value is 99%, the measurements decrease as the camera moves away from the objects. Consequently, the measurements are minimal in the lower-right image in Figure 9. In this scenario, entropy selects the upper-right image in Figure 9 as the best view, likely because the object areas are similar. This view also emerges as the optimal one when the background area comprises 50% of the total area for all three studied measurements. As for Figure 10, the best view is the central right one by giving the background 1% and 50% of the total area, a view that also coincides with the best one for entropy. On the other hand, when giving the background an area of 99% of the total area, the best view is the top left where no objects are observed. These results are consistent with the previous results, highlighting that views can be considered good when the background area represents 50% of the total area. With this in mind, we analyze the following example.



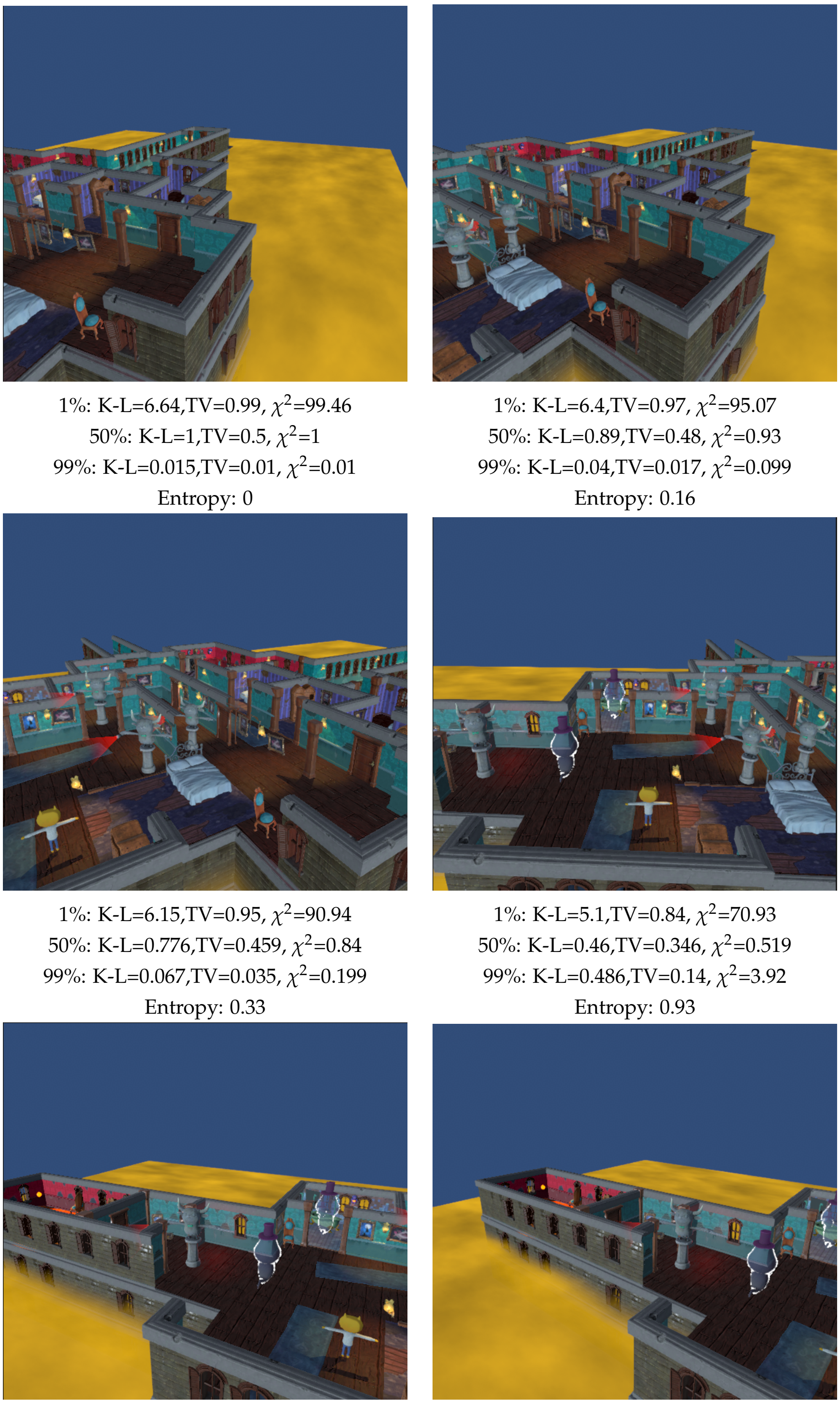
1%: K-L=5.547,TV=0.888, χ^2=79.72
50%: K-L=0.575,TV=0.398, χ^2=0.649
99%: K-L=0.273,TV=0.093, χ^2=1.765
Entropy: 0.69

1%: K-L=4.77,TV=0.796, χ^2=63.965
50%: K-L=0.42,TV=0.32, χ^2=0.455
99%: K-L=0.705,TV=0.182, χ^2=7.123
Entropy: 1.08

1%: K-L=5.55,TV=0.888, χ^2=79.72
50%: K-L=0.575,TV=0.398, χ^2=0.649
99%: K-L=0.273,TV=0.093, χ^2=1.765
Entropy: 0.69

1%: K-L=5.87,TV=0.92, χ^2=85.83
50%: K-L=0.673,TV=0.431, χ^2=0.751
99%: K-L=0.143,TV=0.059, χ^2=0.714
Entropy: 0.5

Figure 9. Comparison of measures (TV: Total Variation, K-L: Kullback–Leibler, χ^2 and Shannon entropy) when zooming out the camera in a videogame scene, and computed with 100,000 rays. The main character is a kitty with a yellow color head, enemies are grey color ghosts and grey color gargoyles with a red torch. Percentages are the relative area assigned to the background.



1%: K-L=6.64,TV=0.99, χ^2=99.46
50%: K-L=1,TV=0.5, χ^2=1
99%: K-L=0.015,TV=0.01, χ^2=0.01
Entropy: 0

1%: K-L=6.4,TV=0.97, χ^2=95.07
50%: K-L=0.89,TV=0.48, χ^2=0.93
99%: K-L=0.04,TV=0.017, χ^2=0.099
Entropy: 0.16

1%: K-L=6.15,TV=0.95, χ^2=90.94
50%: K-L=0.776,TV=0.459, χ^2=0.84
99%: K-L=0.067,TV=0.035, χ^2=0.199
Entropy: 0.33

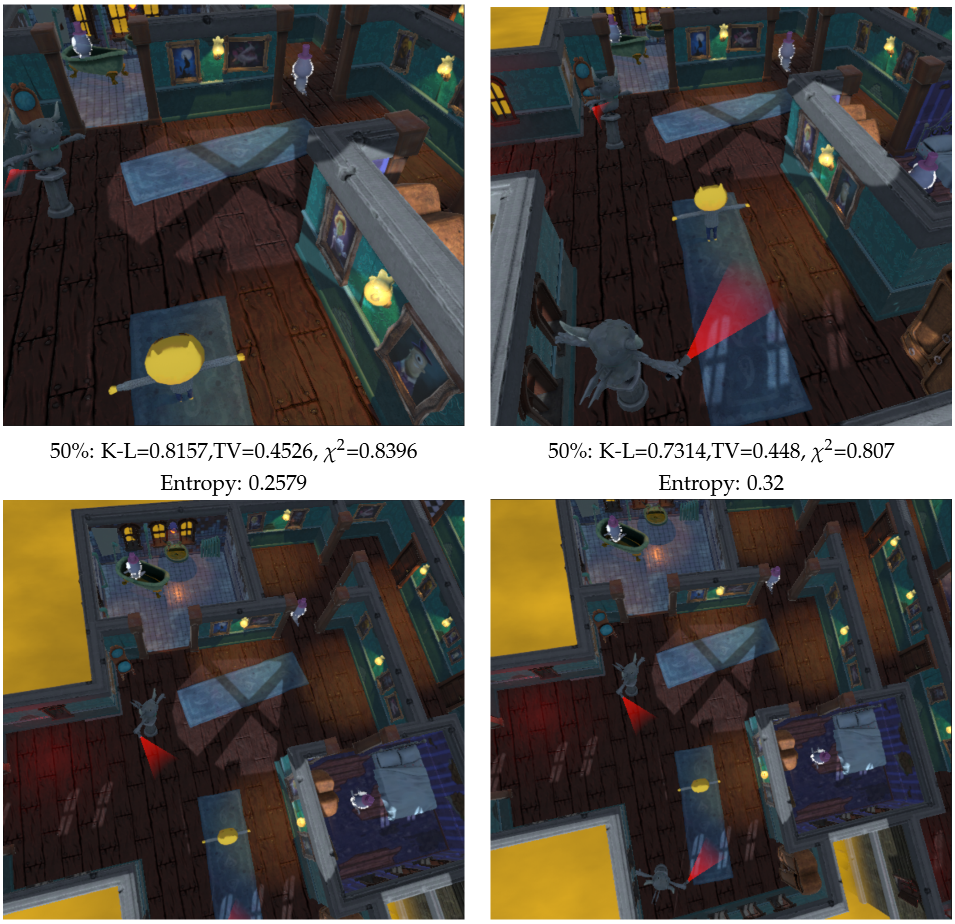
1%: K-L=5.1,TV=0.84, χ^2=70.93
50%: K-L=0.46,TV=0.346, χ^2=0.519
99%: K-L=0.486,TV=0.14, χ^2=3.92
Entropy: 0.93

1%: K-L=5.44,TV=0.866, χ^2=75.85
50%: K-L=0.62,TV=0.378, χ^2=0.62
99%: K-L=0.44,TV=0.12, χ^2=3.85
Entropy: 0.697

1%: K-L=5.69,TV=0.89, χ^2=80.76
50%: K-L=0.69,TV=0.4, χ^2=0.7
99%: K-L=0.33,TV=0.09, χ^2=2.86
Entropy: 0.56

Figure 10. Comparison of measures (TV: Total Variation, K-L: Kullback–Leibler, χ^2 and Shannon entropy) when rotating the camera around a vertical axis in a video game scene, and computed with 100,000 rays. Main character is a kitty with yellow color head, enemies are grey color ghosts and grey color gargoyles with a red torch. Percentages are the relative area assigned to background.

The third evaluated scene is composed of the main character and five enemies, and the rest of the scene is considered as background, see Figure 11. The main character is a kitty with a yellow color head, enemies are grey color ghosts and grey color gargoyles with a red torch. We see two different pairs of rear views, Figure 11 left and Figure 11 right. Figure 11 bottom is zoomed out of Figure 11 top. In Figure 11 left, we cannot appreciate the closest enemy behind the main character, while it is clearly visible in Figure 11 left. Thus, we consider better views than the ones in Figure 11 right.



50%: K-L=0.8157,TV=0.4526, χ^2=0.8396
Entropy: 0.2579

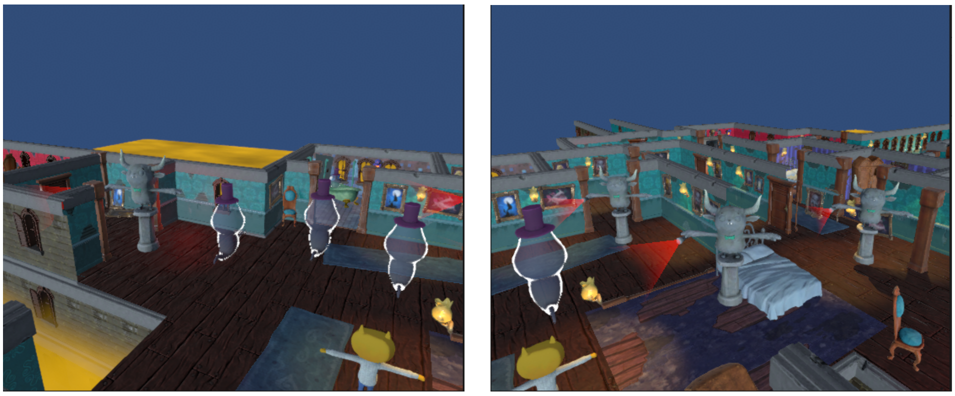
50%: K-L=0.7314,TV=0.448, χ^2=0.807
Entropy: 0.32

50%: K-L=0.9149,TV=0.488, χ^2=0.9532
Entropy: 0.098

50%: K-L=0.9066,TV=0.4877, χ^2=0.9514
Entropy: 0.1059

Figure 11. Comparison of measures in a video game (TV: Total Variation, K-L: Kullback–Leibler, χ^2 and Shannon entropy, percentages are the relative area assigned to background), and computed with 100,000 rays. The main character is a kitty with a yellow color head, enemies are grey color ghosts and grey color gargoyles with a red torch. In the left images, the gargoyle behind main character is not visible.

As another innovative aspect of the implemented method, we consider the importance $\{p_i\}$ of different entities, as described in Section 3.4. Let us envision a scenario in a game with an isometric view positioned behind the main character, featuring a scene populated by eight enemies, four gargoyles and four ghosts. Our objective is to recommend to the player a viewpoint that detects the highest risk. In this case, we deem gargoyles as more offensive than ghosts. Consequently, we assign greater importance to gargoyles. Figure 12 compares two camera views where in Figure 12 left there are three ghosts and one gargoyle, while in Figure 12 right there are three gargoyles and one ghost. Although, at first glance, the view in Figure 12 left might appear preferable due to the enemies being slightly closer, which is reflected in the lower values of the three measures considered, this view contains more ghost-type enemies, which are less aggressive to the player. Therefore, our interest lies in detecting that the most dangerous area is the one in the view in Figure 12 right. However, if we assign greater importance to gargoyles (2, 5, 10 and 20, respectively) while ghosts are given importance 1, we observe that as the gargoyles' importance increases over ghosts one, K-L and χ^2 measures will gradually identify the view in Figure 12 right as superior.



No imp: K-L=0.4201, TV=0.2567, χ^2 =0.4344
imp 2: K-L=0.573, TV=0.2893, χ^2=0.7169
imp 5: K-L=0.8483, TV=0.352, χ^2=1.6111
imp 10: K-L=1.0918, TV=0.3823, χ^2=3.117
imp 20: K-L=1.3519, TV=0.3992, χ^2=6.1346

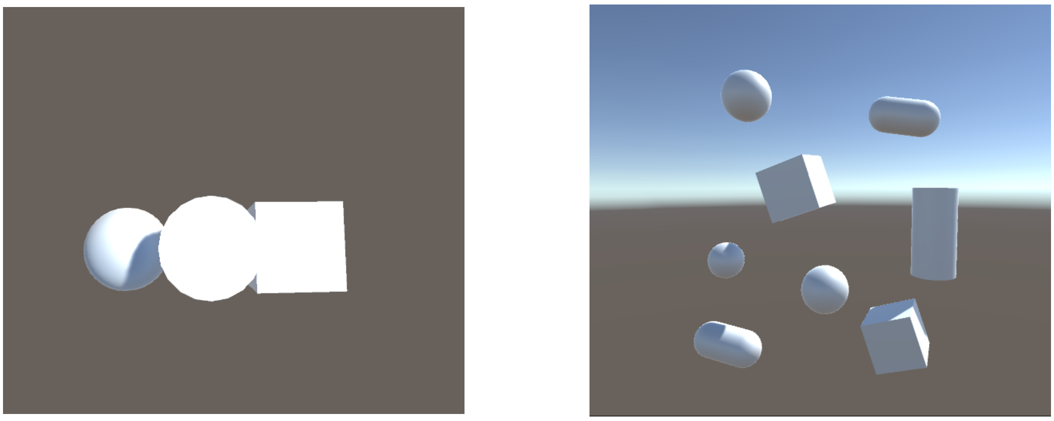
No imp: K-L=0.4461, TV=0.267, χ^2=0.4491
imp 2: K-L=0.3785, TV=0.2436, χ^2=0.3491
imp 5: K-L=0.363, TV=0.2387, χ^2=0.3605
imp 10: K-L=0.3879 ,TV=0.2412, χ^2=0.4832
imp 20: K-L=0.4299, TV=0.2426, χ^2=0.7675

Figure 12. Comparison of measures with importances in a video game (TV: Total Variation, K-L: Kullback–Leibler, and χ^2), computed with 100,000 rays. Background area = 50%. Results in top row are given without importance, while in the second, third, fourth, and fifth row importance values of 2, 5, 10 and 20 are, respectively, assigned to gargoyles and importance 1 to ghosts. There are three ghosts and one gargoyle visible in the left image, while three gargoyles and one ghost in the right image.

4.2. Computation Time

The computation time, obtained from the Time function provided by Unity, comprises a preprocessing time, where areas are calculated and rays are generated, and the time for finding the first hit for each ray and computing the form factors and the measures. The rays are stored and reused for each viewpoint, as the intersections are computed in camera coordinates. With a PC running Windows 10 Pro 64-bit, equipped with an Intel Core i7-6700K CPU @ 4.00 GHz, 16.0 GB RAM, and an NVIDIA GeForce GTX 1080 graphics card using Unity version 2020.3.30f1, the preprocessing time for 100,000 rays is of the order of half a second. In Figure 13 we show the computation time for finding the first hits and computing the measures for different numbers of rays. Time increases proportionally with the number of rays, as expected. The increased cost in the scene in Figure 13 right with respect to the one in Figure 13 left is due to its increased complexity.

The consistency in results suggests that our implementation is robust and scalable, rendering it suitable for real-time applications and interactive environments.



Rays	Time (msec.)
1000	0
10,000	4
100,000	43
1,000,000	434

Rays	Time (msec.)
1000	0
10,000	8
100,000	54
1,000,000	505

Figure 13. Computing time cost (in msec.) in two scenes and with different numbers of rays. The time corresponds to finding first hit for rays and computing the measures. The computation of areas and generation of rays is conducted only once in a preprocessing step, being the cost for 1,000,000 rays being around half a second.

5. Discussion

The Kullback–Leibler divergence is known for its sensitivity to differences in the probability distribution between two sets of data. This makes it especially useful for detecting significant changes in game scenes, as seen for instances in Figure 8, where K-L divergence detects a significant difference between the two scenes while the difference of the values for TV and χ-square divergences and Entropy are within the experimental error threshold. However, K-L divergence can be expensive in terms of calculations in complex scenes.

χ^2 metric stands out for its simplicity and cheaper calculation, which makes it suitable for real-time applications. It is useful to detect differences when areas seen are clearly different, such as in Figures 5 and 6. However, it tends to be less sensitive to subtle differences between distributions compared to K-L as shown in Figure 8.

As for the Total Variation, this is a standardized metric that should facilitate the comparison and interpretation of values in different contexts. This metric, as well as the χ^2 one and different from the K-L one, takes into account the objects that are not seen from the camera, although the possible advantage of TV and χ^2 divergences on K-L divergence is counterbalanced by the use of a background area.

Entropy behaves well in most of our examples, except in Figure 5 upper left, where zooming in on an object increases the entropy and Figure 8 where no improvements are detected. Remember too that the entropy measure considers all objects of equal area and thus the best measure (maximum entropy) would be to see all of them (including background) with an equal form factor, independent of their relative area. On the other hand, divergence measures have the advantage of taking into account the area of the object, and the flexibility of being able to assign importance to them.

All in all, if computation is not an issue and discrimination between measures is a must, K-L divergence can be recommended. If standardization is a must TV divergence can be used, taking care of zooming in on objects. χ^2 divergence represents a balance between good discrimination and cheap computation.

6. Conclusions and Future Work

We have presented in this paper a framework for camera selection in video games that uses information theoretic f-divergences to give the correlation between the visibility from the camera and the objective or target distribution. The visibility is measured by differential area-to-area form factors that are efficiently computed by casting rays using importance sampling Monte Carlo integration. The target by default is the area of the objects but can be modified by assigning importance to them. Thus our approach allows us to take into account the relative importance and preferences of each element in the game. For instance, we can assign higher weights to main characters or key objects to assure higher visual attention for them, in the function of the scene, player characteristics, and game objectives. This can improve the aesthetics of the game and player immersion and experience. The results show the correctness of our approach and seem to favor the K-L divergence as the most discriminating one. We have also shown that the Shannon viewpoint entropy measure is a particular case of K-L divergence when importances are proportional to the inverse of the area.

Currently, our method does not take into account colors or textures for selecting the best view. As part of future work, we plan to include color, illumination and textures, using for instance the importance mechanism. We will also consider the inclusion of the narrative of the game in the camera selection process, as well as more complex game environments, with several kinds of objects and levels of complexity, and evaluate the impact in the improvement of user experience in the different environments.

Another line of work will be to improve the computation time of measures. Using coroutines, or leveraging Unity's job system with the burst compiler to parallelize tasks such as raycasting, can significantly improve performance and responsiveness in complex game environments. Coroutines allow heavy computations to be spread over multiple frames, which reduces frame rate drops and maintains smooth gameplay. Furthermore, the job system and the burst compiler offer a more structured and efficient approach to parallel computing, taking full advantage of multicore processors. Improved computational efficiency is important as we plan to integrate more complex physical properties and narrative elements into the camera selection process, ensuring that these advanced features enhance the gameplay experience.

Reinforcement learning combined with a multi-agent system will be considered too. A machine-learning agent would interact with the game environment, making decisions on camera position and obtaining rewards according to improved visual quality. This approach could contribute to a more customized and narrative-oriented camera selection.

We will also investigate the weighted combination of measures and dynamically adjust the weights according to the kind of objects in the game. This approach could benefit from a multi-agent system, that according to the kind of objects (foes, key elements in the plot or main characters) computes the optimal weights for each metric.

Finally, an extension to 2d games will be considered [46].

Author Contributions: Conceptualization, M.Y.M. and M.S.; Methodology, M.S. and M.C.; Software, M.Y.M.; Validation, M.Y.M., M.S. and M.C.; Resources, M.C.; Data curation, M.Y.M.; Writing—original draft, M.Y.M., M.S. and M.C.; Writing—review & editing, M.Y.M. and M.S.; Visualization, M.Y.M.; Supervision, M.S. and M.C.; Project administration, M.C.; Funding acquisition, M.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been funded by projects CIAICO/2021/037, CIACIF Predoctoral 2023 from Generalitat Valenciana and PDC2021-120997-C31/AEI/10.13039/501100011033 founded by State Investigation Agency from Spanish Government.

Data Availability Statement: The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding authors.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Monte Carlo Computation of Form-Factors

Suppose first, the frustum covers the whole hemisphere. To solve the form-factor hemisphere integral by Monte Carlo integration with importance sampling [7,8,42], let us first consider that ω direction is given by the pair (θ, ϕ), and that $d\omega = \sin\theta d\theta d\phi$.

Then considering as pdf $f(\theta,\phi,x) = \frac{\cos\theta\sin\phi}{\pi}$, and taking N random directions $\{\omega_k = (\theta_k,\phi_k)\}_{k=1}^{N}$ according to $(\theta,\phi) = (\arccos\sqrt{R_1}, 2\pi R_2)$, where R_1 and R_2 are random values from a uniform distribution in the interval [0,1], the value of $F(dA_x,o_i)$ can be estimated as

$$\begin{aligned} F(dA_x,o_i) &\approx \frac{1}{N}\sum_{k=1}^{N}\left(\frac{1}{\pi}\right)\frac{v(x,o_i(\omega_k))\cos\theta_k\sin\theta_k}{f(\theta_k,\phi_k,x_k)} \\ &= \frac{1}{N}\sum_{k=1}^{N}\left(\frac{1}{\pi}\right)\frac{v(x,o_i(\omega_k))\cos\theta_k\sin\theta_k}{\frac{\cos\theta_k\sin\theta_k}{\pi}} \\ &= \frac{1}{N}\sum_{k=1}^{N} v(x,o_i(\omega_k)) = \frac{N_i}{N}, \end{aligned}$$

where $v(x,o_i(\omega_k))$ is a boolean that tells us whether object i is visible from x in direction $\omega_k = (\theta_k,\phi_k)$ and N_i is the number of hits on object i.

Consider now that we restrict the counting of hits on object o_i to the ones that also hit frustum Ω_{fr}. N_i is now the number of rays that hit the frustum and object o_i and N_{fr} is the total number of rays out of the original N cast that hit the frustum. As we only count the first hit, $\sum_i N_i = N_{fr}$. The form-factors $F(dA_x,o_i)$ have to be normalized, thus

$$F(dA_x,o_i) \approx \frac{N_i}{N_{fr}}.$$

References

1. Chen, M.; Feixas, M.; Viola, I.; Bardera, A.; Shen, H.-W.; Sbert, M. *Information Theory Tools for Visualization*; CRC Press: Boca Raton, FL, USA, 2016.
2. Sbert, M.; Feixas, M.; Rigau, J.; Chover, M.; Viola, I. *Information Theory Tools for Computer Graphics*; Synthesis Lectures on Computer Graphics and Animation, Morgan & Claypool; Springer: Berlin/Heidelberg, Germany, 2009.
3. Unity. Available online: https://unity.com/es (accessed on 28 November 2023).
4. Cover, T.M.; Thomas, Joy, A. *Elements of Information Theory*; John Wiley & Sons: Hoboken, NJ, USA, 1991.
5. Sbert, M.; Plemenos, D.; Feixas, M.; Gonzalez, F. Viewpoint quality: Measures and applications. In Proceedings of the First Eurographics Conference on Computational Aesthetics in Graphics, Visualization and Imaging, Girona, Spain, 18–20 May 2005.
6. Bonaventura, X.; Feixas, M.; Sbert, M.; Chuang, L.; Wallraven, C. A survey of viewpoint selection methods for polygonal models. *Entropy* **2016**, *20*, 370. [CrossRef] [PubMed]
7. Sbert, M. An Integral Geometry Based Method for Fast Form Factor Computation. *Comput. Graph. Forum* **1993**, *12*, 409–420. [CrossRef]
8. Sbert, M. The Use of Global Directions to Compute Radiosity: Global Monte Carlo Techniques. Ph.D. Thesis, Catalan Technical University, Barcelona, Spain, 1997.
9. Siegel, R.; Howell, J.R. *Thermal Radiation Heat Transfer*; Taylor & Francis: Abingdon-on-Thames, UK, 1992.
10. Cohen, M.F.; Wallace, J.R. *Radiosity and Realistic Image Synthesis*; The Morgan Kaufmann Series in Computer Graphics; Morgan Kaufmann: Burlington, MA, USA, 1981.
11. Sillion, F.X.; Puech, C. *Radiosity and Global Illumination*; Morgan Kaufmann: Burlington, MA, USA, 1994.
12. Dutre, P.; Bekaert, P.; Bala, K. *Advanced Global Illumination*, 1st ed.; A K Peters: Natick, MA, USA; CRC Press: Boca Raton, FL, USA, 2003.
13. Plemenos, D.; Benayada, M. Intelligent display techniques in scene modelling. New techniques to automatically compute good views. In Proceedings of the International Conference GraphiCon'96, St Petersbourg, Russia, 1 July 1996.
14. Vázquez, P.; Feixas, M.; Sbert, M.; Heidrich W. Viewpoint Selection Using Viewpoint Entropy. In Proceedings of the Vision Modeling & Visualization Conference, Stuttgart, Germany, 21–23 November 2001.
15. Andújar, C.; Vázquez, P.; Fairén, M. Way-finder: Guided tours through complex walkthrough models. In *Computer Graphics Forum*; Blackwell Publishing, Inc.: Oxford, UK, 2004; Volume 23, pp. 499–508.
16. Zeng, R.; Wen, Y.; Zhao, W.; Liu, Y.J. View planning in robot active vision: A survey of systems, algorithms, and applications. *Comput. Vis. Media* **2020**, *6*, 225–245. [CrossRef]
17. Bordoloi, U.D.; Shen, H.W. View selection for volume rendering. In Proceedings of the VIS 05. IEEE Visualization, 2005, Minneapolis, MN, USA, 23–28 October 2005.
18. Polonsky, O.; Patané, G.; Biasotti, S.; Gotsman, C.; Spagnuolo, M. What's in an image? Towards the computation of the "best" view of an object. *Vis. Comput.* **2005**, *21*, 840–847. [CrossRef]
19. Secord, A.; Lu, J.; Finkelstein, A.; Singh, M.; Nealen, A. Perceptual models of viewpoint preference. *ACM Trans. Graph.* **2011**, *30*, 5. [CrossRef]

20. Stoev, S.L.; Straßer W. A case study on automatic camera placement and motion for visualizing historical data. In Proceedings of the IEEE Visualization 2002, Boston, MA, USA, 27 October–1 November 2002.
21. Feixas, M.; Sbert, M.; Gonzalez, F. A unified information-theoretic framework for viewpoint selection and mesh saliency. *ACM Trans. Appl. Percept.* **2008**, *6*, 1. [CrossRef]
22. Vázquez, P.P. Automatic view selection through depth-based view stability analysis. *Vis. Comput.* **2009**, *25*, 441–449. [CrossRef]
23. Delmerico, J.; Isler, S.; Sabzevari, R.; Scaramuzza, D. A comparison of volumetric information gain metrics for active 3D object reconstruction. *Auton Robot.* **2018**, *42*, 197–208. [CrossRef]
24. Zhang, Y.; Fei, G. Overview of 3D scene viewpoints evaluation method. *Virtual Real. Intell. Hardw.* **2019**, *1*, 341–385. [CrossRef]
25. Sokolov, D.; Plemenos, D.; Tamine, K. Methods and data structures for virtual world exploration. *Vis. Comput.* **2006**, *22*, 506–516. [CrossRef]
26. Lee, C.H.; Varshney, A.; Jacobs, D.W. Mesh saliency. In Proceedings of the SIGGRAPH05: Special Interest Group on Computer Graphics and Interactive Techniques Conference, Los Angeles, CA, USA, 31 July–4 August 2005.
27. Attene, M.; Katz, S.; Mortara, M.; Patane, G.; Spagnuolo, M.; Tal, A. Mesh segmentation—A comparative study. In Proceedings of the IEEE International Conference on Shape Modeling and Applications, Matsushima, Japan, 14–16 June 2006.
28. Takahashi, S.; Fujishiro, I.; Takeshima, Y.; Nishita, T. A feature-driven approach to locating optimal viewpoints for volume visualization. In Proceedings of the IEEE Conference on Visualization, Minneapolis, MN, USA, 23–28 October 2005.
29. Ruiz, M.; Bardera, A.; Boada, I.; Viola, I.; Feixas, M.; Sbert, M. Automatic Transfer Functions Based on Informational Divergence. *IEEE Trans. Vis. Comput. Graph.* **2011**, *17*, 1932–1941. [CrossRef] [PubMed]
30. Lan, K.; Sekiyama, K. Optimal viewpoint selection based on aesthetic composition evaluation using Kullback–Leibler divergence. In Proceedings of the Intelligent Robotics and Applications: 9th International Conference, ICIRA 2016, Tokyo, Japan, 22–24 August 2016.
31. Yokomatsu, T.; Sekiyama, K. Optimal Viewpoint Selection by Indoor Drone Using PSO and Gaussian Process With Photographic Composition Based on K-L Divergence. *IEEE Access* **2022**, *10*, 69972–69980. [CrossRef]
32. Galvane, Q. Automatic Cinematography and Editing in Virtual Environments. Artificial Intelligence. Ph.D. Thesis, Université de Grenoble Alpes, Grenoble, France, 2015.
33. Lino, C.; Christie, M. Efficient composition for virtual camera control. In Proceedings of the ACM SIGGRAPH Symposium on Computer Animation, Los Angeles, CA, USA, 5–9 August 2012.
34. Zhang, Y.; Fei, G.; Gang, Y. 3D viewpoint estimation based on aesthetics. *IEEE Access* **2020**, *8*, 108602–108621 [CrossRef]
35. Kiciroglu, S.; Rhodin, H.; Sinha, S.N.; Salzmann, M.; Fua, P. Activemocap: Optimized viewpoint selection for active human motion capture. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
36. Hartwig, S.; Schelling, M.; Onzenoodt, C.V.; Vázquez, P.P.; Hermosilla, P.; Ropinski, T. Learning human viewpoint preferences from sparsely annotated models. *Comput. Graph. Forum* **2022**, *41*, 453–466. [CrossRef]
37. Polyanskiy, Y.; Wu, Y. *Information Theory: From Coding to Learning*; Cambridge University Press: Cambridge, UK, 2022.
38. Wikipedia. F-Divergence—Wikipedia, The Free Encyclopedia, Wikipedia, The Free Encyclopedia. Available online: http://en.wikipedia.org/w/index.php?title=F-divergence&oldid=1068442350 (accessed on 12 October 2023).
39. Sason, I. On Data-Processing and Majorization Inequalities for f-Divergences with Applications. *Entropy* **2019**, *21*, 1022. [CrossRef]
40. Sbert, M.; Szirmay-Kalos, L. Robust Multiple Importance Sampling with Tsallis ϕ-Divergences. *Entropy* **2022**, *24*, 1240. [CrossRef]
41. Rigau, J.; Feixas, M.; Sbert, M. *Information Theory Point Measures in a Scene*; IIiA-00-08-RR, Institut d'Informàtica i Aplicacions, Universitat de Girona: Girona, Spain, 2000.
42. Rubinstein, R.Y. *Simulation and the Monte Carlo Method*; Wiley Series in Probability and Statistics; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 1981.
43. Cohen, M.F.; Greenberg, D.P. The hemi-cube: A radiosity solution for complex environments. *ACM SIGGRAPH Comput. Graph.* **1985**, *19*, 31–40. [CrossRef]
44. Sillion, F.; Puech, C. A general two-pass method integrating specular and diffuse reflection. *ACM SIGGRAPH Comput. Graph.* **1989**, *23*, 335–344. [CrossRef]
45. John Lemon's Haunted Jaunt: 3D Beginner—Unity Learn. Available online: https://learn.unity.com/project/john-lemon-s-haunted-jaunt-3d-beginner (accessed on 28 November 2023).
46. Orti, R.; Riviere, S.; Durand, F.; Puech, C. Radiosity for dynamic scenes in flatland with the visibility complex. *Comput. Graph. Forum* **1996**, *15*, 237–248. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.