International Conference on Artificial Reality and Telexistence
Eurographics Symposium on Virtual Environments (2015)
M. Imura, P. Figueroa, and B. Mohler (Editors)

Comparison and Evaluation of Viewpoint Quality Estimation Algorithms for Immersive Virtual Environments

S. Freitag[†,1], B. Weyers [1], A. Bönsch [1], and T. W. Kuhlen [1,2]

[1] Virtual Reality Group, RWTH Aachen University — JARA – High-Performance Computing
[2] Jülich Supercomputing Centre

Abstract

The knowledge of which places in a virtual environment are interesting or informative can be used to improve user interfaces and to create virtual tours. Viewpoint Quality Estimation algorithms approximate this information by calculating quality scores for viewpoints. However, even though several such algorithms exist and have also been used, e.g., in virtual tour generation, they have never been comparatively evaluated on virtual scenes.
In this work, we introduce three new Viewpoint Quality Estimation algorithms, and compare them against each other and six existing metrics, by applying them to two different virtual scenes. Furthermore, we conducted a user study to obtain a quantitative evaluation of viewpoint quality. The results reveal strengths and limitations of the metrics on actual scenes, and provide recommendations on which algorithms to use for real applications.

Categories and Subject Descriptors (according to ACM CCS): H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Evaluation/methodologies I.2.10 [Artificial Intelligence]: Vision and Scene Understanding—3D/stereo scene analysis

1. Introduction

Increasing user efficiency and effectiveness in immersive virtual environments (IVEs) is an important task that often involves the automation of otherwise user-controlled parameters or degrees of freedom (e.g., travel speed [MMGK09] or selection assistance [DHKP05]). The success of these methods usually depends on the accuracy of the system's prediction of what a user wants to do, or where she wants to do it. Of course, both strongly depend on the user's current goal which the system does not necessarily know. However, in many tasks such as scene exploration—which is usually performed by all users for each new scene—good positions are often those that help the user get to know the environment or gain information about its state. As this is usually done visually, a good position or *viewpoint* is often one that offers a large amount of accessible (i.e., visible) information.

To determine good viewpoints, several Viewpoint Quality Estimation (VQE) algorithms already exist. In most cases, they target single 3D objects seen from the outside, for example for determining good initial positions in 3D editing software or automatically generating 2D previews. These algorithms are based on various principles, ranging from simple silhouette length or visible area [PPB*05] over information entropy (e.g., [VFSH03, SPFG05]) to mesh curvature (e.g., [PKS*03, LVJ05]) and can also be based on 3D interest point detection techniques [DCG12]. Overviews and comparisons of many of these approaches, applied to 3D objects, can be found in [DCG10], [DCG12] and [SPFG05], nine algorithms are also introduced in more detail in section 2.1.

Although most of these measures target single 3D objects, there are several approaches that apply them to scenes, for example for automatic scene exploration (e.g., [BDP00, AVF04, VFSH03, JTP06, SP05]). However, the question remains if these methods can perform better using different VQE metrics, or algorithms specifically designed for architectural scenes. Nevertheless, to the best of our knowledge, there has not been a comparison, evaluation or discussion of the performance of different VQE algorithms for 3D scenes, nor any automatic metric that makes use of scene-specific properties.

Therefore, the main contributions of this work are:

- We introduce three new VQE metrics for scenes (*object area entropy*, *relative object area entropy* and *object uniqueness*), two of which are adaptations of existing techniques, while one is a completely new approach.

† freitag@vr.rwth-aachen.de

DOI: 10.2312/egve.20151310

- We compare these, and six existing metrics, applied to two different architectural scene models (a house and an office floor) and discuss their strengths and limitations.
- We evaluate the performance of the nine algorithms based on human ground truth determined in a user study.
- We discuss applications for VQE algorithms for scenes.

The rest of the paper is structured as follows. Section 2 discusses the main differences between VQE for single 3D objects and architectural scenes, and introduces the nine algorithms that are compared in this paper. Section 3 presents the results of all algorithms on both scenes and a discussion of their strengths and limitations. The user study we conducted to acquire human ground truth data is introduced and discussed in section 4. Section 5 gives a short overview of useful applications of VQE metrics, before the paper is concluded in section 6 along with an outlook on future work. Raw data, visualizations and more information about the scenes can be found in [FWBK15].

2. Viewpoint Quality in Architectural Scenes

The task of VQE in architectural scenes (usually viewed inside-out) is different from the same task for single 3D objects (mostly viewed outside-in), even though the same algorithms can often be applied to both (e.g., [VFSH03]). Maybe the most prominent distinction arises from self-occlusion problems: in architectural scenes, most details remain inaccessible if only outside viewpoints are available, while self-occlusion is usually less of a problem for single geometric objects. Furthermore, for outside-in viewpoints, most details are usually found in a small set of view directions, while potentially all view directions of inside-out viewpoints contain valuable information. The set of possible viewpoints is also very similar for different objects viewed outside-in (e.g., a sphere around the object), while for inside-out views it strongly depends on the actual scene. Finally, most scenes can be decomposed into meaningful objects they consist of, which is harder or impossible to do automatically for most single objects.

2.1. Algorithms

For comparison, we selected six algorithms that have been used for best automatic viewpoint selection in different publications and can be—or already have been—used in architectural scenes. In addition, we introduce three new metrics that are based on the concept of *objects* that can be identified in scenes.

Using the number of visible objects as criterion for viewpoint quality has already been suggested [SP05] and applied in a simple heuristic [JTP06]. However, objects had to be defined manually, which is a strong restriction for complex scenes. In contrast, our three new techniques use the fact that almost all scenes are divided into geometries or groups anyway by their modelers, which is also preserved in most file formats. Even though these groups do not necessarily correspond to the human perception of an "object", they are presumably a good approximation if made by a human. Furthermore, object grouping in human perception is not unambiguous (e.g., is a tree an object, or are all its leaves separate objects?). To avoid considering several (semantic) objects as one, we extract an object for each lowest-level group, even though humans would probably often summarize a few of them into a single object. Note that a more meaningful object extraction may lead to better results, but is not trivial to compute.

We implemented all algorithms as in their original source (unless indicated otherwise). For visibility detection of, e.g., triangles or vertices, we use the GPU. To avoid direction-dependent perspective distortion, the scene should be projected onto a sphere. We simulate this by rendering a cube map and correcting the pixel areas in subsequent calculations where necessary (similar to [VFSH03]), except in *depth map stability* where this is not easily possible. Unless otherwise noted, all cube map sides are rendered with a resolution of 1024×1024 pixels. The visibility and projected area of vertices, polygons and objects is determined by using an item buffer (drawing each entity in a different color) and summing over the projected area of each color on the projection sphere.

In the following, we give a short overview over all algorithms that we compared, along with our implementation.

Surface area entropy (also called *viewpoint entropy*) interprets the projected area of each polygon relative to the total area projected on the projection sphere with respect to a viewpoint as a probability [VFSH01, VFSH03]. The entropy of this probability distribution is calculated as $SAE = -\sum_{i=1}^{N_f} \frac{a_i}{a_t} \log_2(\frac{a_i}{a_t})$, where N_f is the number of faces, a_i the projected area of the i-th polygon, and $a_t = \sum_{i=1}^{N_f} a_i$ the total projected area over the projection sphere (ignoring the background). It reaches its maximum $\log(N_f)$ when all polygons get the same projected area.

Relative surface area entropy is similar to *surface area entropy*. It is defined as the Kullback-Leibler (KL) distance [CT12] between the probability distribution of projected polygon areas and the probability distribution of their actual areas [SPFG05]. It is calculated as $RSAE = \sum_{i=1}^{N_f} \frac{a_i}{a_t} \log_2(\frac{a_i}{a_t} / \frac{A_i}{A_T})$, where A_i is the actual area of polygon i and A_T the total surface area of all polygons. As this is a distance measure, we use $-RSAE$ as the viewpoint quality.

Ratio of visible area is defined as the ratio between the visible and the total 3D surface area [PPB*05] that we approximate using the GPU by evaluating the depth buffer.

Curvature entropy interprets the curvature of a vertex v as a probability. It is defined as the entropy over the normalized probability distribution of the visible vertex curvatures $\overline{\mathscr{C}(v)}$: $CE = -\int_{-\infty}^{\infty} \overline{\mathscr{C}(v)} \log_2 \overline{\mathscr{C}(v)} dv$ [PKS*03]. We implemented it by calculating mean curvatures for each vertex using Taubin's method [Tau95], and discretizing the curvature values of visible vertices into a histogram. CE is maximal when all curvature values appear equally often from a viewpoint.

Mesh saliency is inspired by Itti's saliency model for intensities [IKN98]. It applies a center-surround mechanism to vertex mean curvatures to identify regions that differ from

their surrounding context, combining several scales for robust detection of salient features. We implemented *mesh saliency* as in [LVJ05], computing mean curvatures $\mathscr{C}(v)$ for each vertex v using Taubin's method [Tau95], Gaussian-weighted averages of mean curvatures $G(\mathscr{C}(v),\sigma)$ for a scale σ, and the saliency map at that scale as $\mathscr{S}(v,\sigma)=|G(\mathscr{C}(v),\sigma)-G(\mathscr{C}(v),2\sigma)|$. As in [LVJ05], this is combined for the scales $\sigma\in\{2\varepsilon,3\varepsilon,4\varepsilon,5\varepsilon,6\varepsilon\}$ (where $\varepsilon=0.3\%$ of the diagonal of the scene's bounding box) using Itti's non-linear normalization operator [IKN98] that promotes maps with few peaks and suppresses those with many similar peaks.

Depth map stability (also called *depth-based view stability*) is inspired by the observation that good views on objects are often geometrically stable, i.e., stable regarding their depth values [Váz09]. The stability of a viewpoint is calculated as the mutual information (similarity) of its depth map and those of all other examined viewpoints. As in [Váz09], we implement this by using the *bzip2* algorithm. As *bzip2* compresses blocks only up to a size of 900KB, the depth map size is restricted to a maximum of 273×273 pixels (8bit resolution), as 12 views have to be compressed in the same block for two cube maps (we used 256×256 pixels). As VQE, the sum of similarities between a viewpoint and all other viewpoints is used in [Váz09]. However, as this requires $O(n^2)$ comparisons for n viewpoints, we only compared each viewpoint to $\approx$350 viewpoints sampled in a regular grid over the scene at the same eye height.

Object area entropy is our adaptation of *surface area entropy* to objects, calculated the same way, only exchanging polygons for objects. Thus, the projected polygon areas a_i and a_t are replaced by the projected object areas α_i and α_t.

Relative object area entropy is our adaptation of *relative surface area entropy* to objects. It is calculated identically, except that projected polygon areas a_i and a_t are replaced by projected object areas α_i and α_t, and actual polygon areas A_i and A_T by actual object areas (mesh surface) $\mathcal{A}_i$ and $\mathcal{A}_T$.

Object uniqueness is a new measure, motivated by the observation that objects that are visually distinctive and unique in a given environment also tend to convey more information to a viewer than objects that occur repeatedly. For each object i, a *uniqueness value* $U(i)\in(0,1]$ is computed that captures its uniqueness regarding geometry and color (as described below), where a value of 1 indicates a totally unique object. The uniqueness score $US\in[0,1]$ of a view is computed based on the uniqueness of the visible objects, and is maximal when $\forall i:\alpha_i\propto U(i)$, i.e., when the projected area $\alpha(i)$ of all objects is proportional to their uniqueness value $U(i)$.

To compare object appearances, descriptors for the geometries and color distribution of all objects are computed. The geometry descriptor has to be invariant to translation and rotation (object placement). Furthermore, we argue that it should also be invariant to uniform scaling. Although, e.g., a giant teacup is probably quite different from a regular-sized one, in most scenes probably only small size differences occur. We also decided to use a descriptor invariant to non-uniform scaling, to have, e.g., books or boxes of different sizes rated as similar. We chose to use Osada's D2 descriptor [OFCD01] that represents the distribution of the distances between two points on the surface of an object and is calculated by repeatedly sampling two random points on the surface and determining their distance. D2 is invariant to rotation, translation, and uniform scaling. To make it invariant to non-uniform scaling, we perform a PCA on the object's geometry to identify its main axes and then scale it along these axes to unit length. As color descriptor, we compute a color histogram (including texture), using the mostly perceptually uniform CIELAB color space. The final descriptor $D(i)$ of object i is a histogram with 472 bins: 256 bins for D2 and 216 bins for the color histogram. It has to be noted, though, that the quality of the object uniqueness values heavily depends on the object descriptor used. Other descriptors (such as PFHRGB or SHOTCOLOR [RC11]) could yield better results [Ale12].

We then compute similarities between the descriptors of all objects. Let $BC(s,t)$ be the Bhattacharyya coefficient [Bha43] between two normalized histograms s and t, i.e., $BC(s,t)=\sum_i\sqrt{s_i\cdot t_i}$, that measures the amount of overlap between s and t. It is easy to interpret, as $BC(s,t)=0$ when s and t do not overlap and $BC(s,t)=1$ when $s=t$. As only very similar descriptors indicate actually similar objects, we only treat objects with a high descriptor similarity as "similar". Therefore, let the similarity between two objects i and j be $S(D(i),D(j))=\max(0,\frac{BC(D(i),D(j))-\theta}{1-\theta})$ with θ close to 1 (we use $\theta=0.975$). The uniqueness $U(i)$ of an object i is then defined as the reciprocal of the sums of all similarity values, i.e., $U(i)=(\sum_j S(D(i),D(j)))^{-1}$. The uniqueness value of an object i that appears identically m times, and is dissimilar to all other objects, is thus $U(i)=\frac{1}{m}$. A totally unique object j (not similar to anything) has $U(j)=1$. The uniqueness score US of a view is then calculated as the Bhattacharyya coefficient between the normalized probability distributions of the projected areas of all objects and their uniqueness values, i.e., $US=\sum_i\sqrt{\frac{\alpha_i}{\alpha_t}\cdot\frac{U(i)}{U_T}}$, where $U_T=\sum_i U(i)$.

We expect this score to be higher for viewpoints that are closer to small, unique objects (that are presumably more important), and farther away from non-unique (presumably less interesting) objects as well as large unique objects that can be comfortably seen at a distance.

3. Algorithm Comparison

Which VQE algorithm provides the best results on a given scene depends on the application and the desired properties of the result. For example, when selecting best viewpoints, it is essential that (global or local) maxima are located correctly, while it is less important whether very bad viewpoints get a very low or a medium score. However, when choosing travel speeds based on viewpoint quality (see section 5), the actual scores are important everywhere. Therefore, in this section, we will not argue which algorithm performs best, but discuss their strengths and limitations in certain circumstances.

3.1. Scenes

There are various classes of architectural scenes, e.g., indoor, outdoor and mixed scenes, small rooms or large cathedrals and highly detailed or sparsely modeled environments, and it can be expected that different algorithms show their strengths in different kinds of scenes. For the scope of this work, we focus on indoor scenes (cf. Figure 1) for the following reasons. First, outdoor scenes tend to have a wide variation, with natural as well as man-made structures of various sizes—making it harder to extrapolate from few scenes—while indoor areas are usually more similar in structure. Second, in indoor scenes, realistic viewpoints are usually at eye height, while it can make sense to fly in outdoor scenes, making the analysis more application-dependent. To cover a large amount of variation, we selected two scenes of different types. The first one is a small, detailed house, with furnished rooms and many different objects, especially in the kitchen (kitchen appliances and tools) and the living room (books, DVDs, a laptop, etc.). The second one is a larger, less-detailed office floor with similar interiors for many rooms, as well as mostly empty corridors. It also contains some closed rooms without content, as is often the case for irrelevant areas to save modeling effort. In both scenes, the ground is at the same height everywhere. Furthermore, only little information is contained in textures, and there are no (semi-)transparent objects, as only one algorithm under comparison (object uniqueness) respects color or texture, and no algorithm accounts for transparency.

For both scenes, we calculated the score of all algorithms with a high spatial resolution, using a regular 0.05 m (house) or 0.10 m (office) grid. As we focus on realistic viewpoints for (virtual) human observers, they were selected at eye height above the ground, accounting for different eye heights by averaging the score over heights from 1.45 m to 1.80 m in 0.05 m increments. In total, a little over 1 million viewpoints were evaluated by each algorithm for each scene.

3.2. Results and Discussion

The results of the algorithm scores for both scenes are visualized in Figure 2 as heat maps. The heat maps show the normalized viewpoint score, using a linear color mapping from black (min) over red and yellow to white (max) in equal intervals.

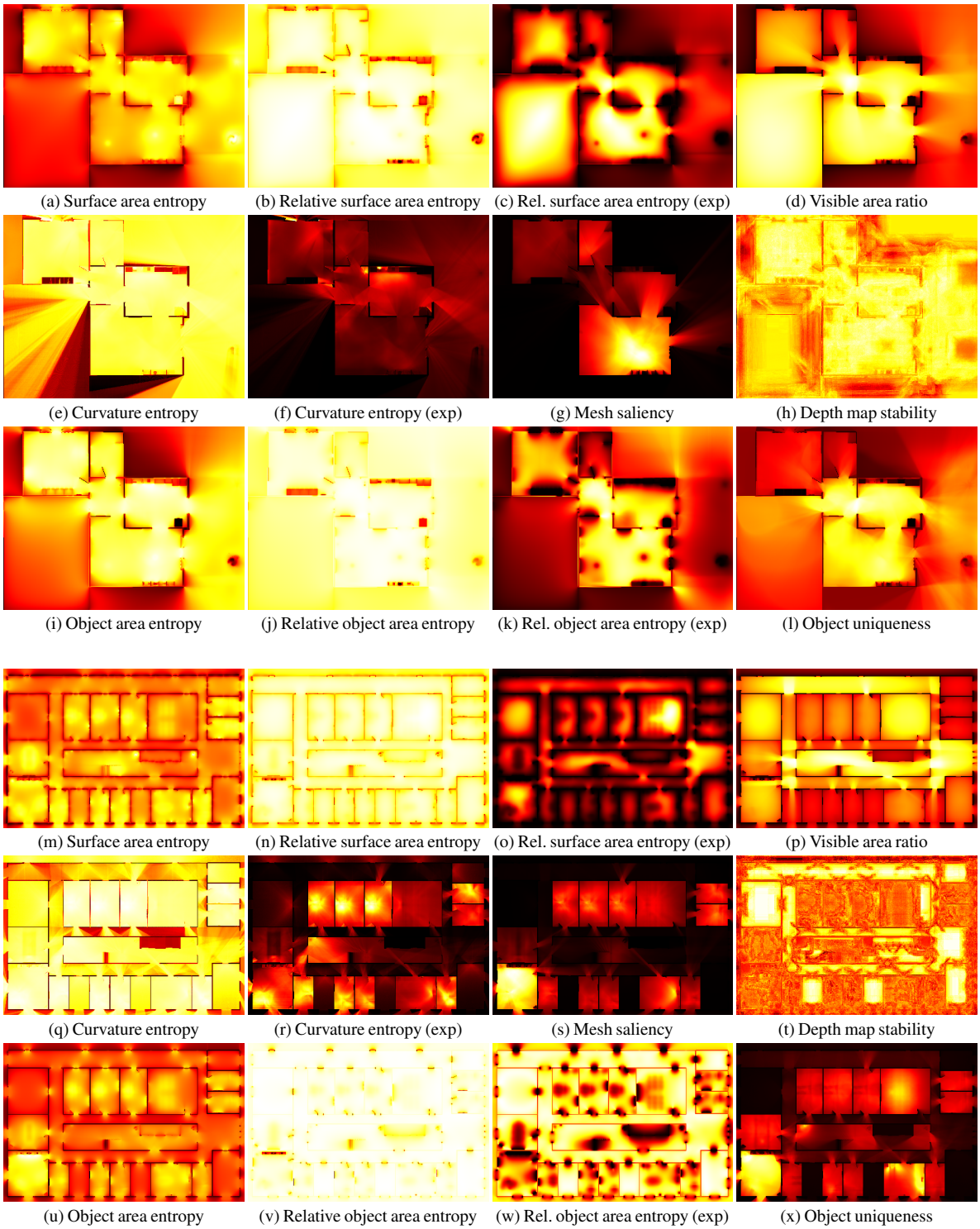
Surface area entropy is attracted to highly tessellated areas and rates views higher when more triangles are visible, and when triangle sizes are more equally distributed. While highly tessellated regions often correspond to more information-rich areas—e.g., regions in the house and in furnished offices are generally rated higher than empty regions—this often leads to undesired behavior. For example, viewpoints in the (high-tessellated) fridge and on both sides of the crinkled shower curtain (house) or in the potted plants (patio/central office corridor) receive very high scores, even though the view itself is probably uninformative to a human observer. When selecting representative views from high scores, this is most likely unwanted behavior. In the office, all furnished rooms receive similarly high scores. The rooms' maxima, however, are usually close to pinboards that hold some tesselated papers and pins.

Figure 1: Top view of the house and office scenes. High-res versions and an overview video can be found in [FWBK15].

Relative surface area entropy rates views higher when the projected size of polygons is proportional to their actual sizes. In rooms, this often leads to central overview viewpoints receiving higher scores, as more triangles are visible and viewed from less inclined angles. However, very large polygons also get a large weight, leading to high scores in the center of the empty outside area of the house, and empty rooms and corridors in the office. The large difference between the scores of very bad and most other viewpoints is caused by the fact that the KL distance can get very large for bad viewpoints.

Visible area ratio always rates views in larger rooms higher (as more surface is visible). Therefore, the empty outside area in the house and the corridors and large, empty rooms in the office all get high scores, which is probably undesired. Within the house, the measure favors central viewpoints in sight of doorways (where another room can be seen), which might correspond to informative viewpoints.

Curvature entropy is based on the visibility of vertices and therefore has less smooth transitions between neighboring viewpoints than algorithms based on objects or polygons that can also be partially seen. The method rates empty rooms in



(a) Surface area entropy (b) Relative surface area entropy (c) Rel. surface area entropy (exp) (d) Visible area ratio

(e) Curvature entropy (f) Curvature entropy (exp) (g) Mesh saliency (h) Depth map stability

(i) Object area entropy (j) Relative object area entropy (k) Rel. object area entropy (exp) (l) Object uniqueness

(m) Surface area entropy (n) Relative surface area entropy (o) Rel. surface area entropy (exp) (p) Visible area ratio

(q) Curvature entropy (r) Curvature entropy (exp) (s) Mesh saliency (t) Depth map stability

(u) Object area entropy (v) Relative object area entropy (w) Rel. object area entropy (exp) (x) Object uniqueness

Figure 2: Algorithm results for the house (a–l) and the office (m–x), visualized as heat maps (normalized to the range [0,1]). The color mapping is a linear gradient from black for the worst value (0), over red ($1/3$) and yellow ($2/3$) to white (1). For some logarithmically scaled algorithms, heat maps of exponentiated scores (2^{score}) are shown as well for better distinguishability (marked with (exp)). Raw images and data, including exponentiated versions of *surface* and *object area entropy* (a, i, m, u) can be found in [FWBK15].

the office consistently low due to the low number of visible vertices (with different curvatures) and gives higher scores to rooms with content. The high scores are always awarded to viewpoints above desks, as then more (different) vertices of highly resolved models can be seen. In the house, the highest scores are all at clearly unfavorable positions: behind a kitchen appliance, behind a bedroom curtain, and behind the clothes hanger in the hall.

Mesh saliency is also based on vertex visibility, and like *curvature entropy* gives low scores to viewpoints with low numbers of visible vertices, therefore successfully excluding empty rooms and uninformative outside areas. In the house, the highest-rated point by far is in the living room above the table, due to the high diversity of details. While this is probably an informative viewpoint, all other points in the house that are not directly in its view probably receive too low scores in comparison. In the office, the evaluation seems mostly intuitive, with the highest scores by far in the (diverse) boss' office. However, most other offices and especially the meeting room and central corridor receive low or very low scores.

Depth map stability gives high scores to viewpoints with depth profiles similar to that of many other viewpoints. Most of the highest scores in the house are therefore in rooms due to the common depth profile of the walls. In the office, however, the highest scores can all be found in empty rooms and corridors. In addition, the results are very noisy, and medium or low scores can often be found close to high scores.

Object area entropy works similar to *surface area entropy*, but as it is independent of tessellation, it avoids most of its drawbacks. For example, viewpoints in the fridge, shower curtain and potted plants all get low scores. The result in the office is similar to that of *surface area entropy*, with generally lower scores on the (high-poly) stair cordon and the empty regions, and higher scores in the boss' office.

Relative object area entropy has a similar result as *relative surface area entropy*. However, as it is independent of tessellation, the outside area in front of the house (containing large triangles) gets a much lower score. Although it also contains large objects (e.g., walls) that receive a high score when projected on a large area, these can also be seen from within the building (in addition to many other objects). In the office, however, the result is worse: as the object surface areas are dominated by huge objects (all walls are recognized as a single object), the highest scores are achieved where these get most projection surface, i.e., in empty rooms. This behavior is similar to *relative surface area entropy* for very large polygons, but can be more extreme when large parts of the scene are recognized as one object.

Object uniqueness reaches reasonable results for the house scene, with high scores distributed throughout the kitchen, living room and hall (from which many rooms can be seen). Viewpoints in enclosed areas (e.g., cupboards) consistently get very low scores. The best viewpoint is in the doorway between hall and kitchen, a point from where most of the kitchen, but also parts of the living room, bedroom and bathroom can be seen. However, giving similar scores to the empty outside area as to bedroom and bathroom is probably undesired. This is likely caused by both rooms containing only few objects (some of which are repeated), while from the outside area, the large and unique building blocks of the house (and parts of the hall) are visible. In the office, the method produces intuitive results, with very low scores in empty offices and low scores in empty corridors. The regular offices with very similar setup—where seeing only one in detail is probably sufficient—receive medium scores, while the more unique rooms (seminar room, kitchen, boss' office) get higher scores.

4. Study

We conducted a user study in a CAVE to quantitatively evaluate the algorithm results against human ground truth data. It consisted of three parts: an exploration and two evaluation parts (one for each scene). In the first part, participants were asked to explore each scene thoroughly for 10 minutes. They were told that afterwards, they would be asked questions regarding any information about both scenes, and to prepare for these as well as they could. As incentive, the three participants who performed best on the questions received a free lunch or drinks.

In the evaluation parts, participants were teleported to a series of viewpoints in random order. They rated each one with a score between 0 and 4 using five buttons on an input device before they were moved to the next one. They were told to use the lowest score 0 for the most uninformative viewpoint(s) in the scene, and the highest score 4 for the most informative one(s). To include possibly different concepts of informativeness (of different people) in the rating, participants were not given concrete examples of informative viewpoints. To avoid bias, the viewpoints were chosen by a regular sampling, using a 1 m grid in the house scene, and 3 m in the office. Furthermore, in the house scene, only viewpoints in the house and on the patio were considered to reduce the total number. In total, participants saw 159 viewpoints in the house and 176 in the office (which we found to be a high, but manageable number), and took 16 minutes to complete each scene on average. They were allowed to physically turn around 360°, but not to move away or crouch.

The three study parts were conducted on three different dates to ensure that participants could remain concentrated during the (rather repetitive) evaluation. The study took place in a five-sided CAVE (4 walls + floor), which means that participants could look around, but not see what was directly above them. However, as almost no details (except for lamps) were located above, we assume that this did not influence the result in a significant way. In total, 22 people (2 female, mean age 28.2) participated in the study.

4.1. Results

All user evaluations were averaged for each viewpoint (cf. Figure 3). For each VQE algorithm, we calculated the score for each viewpoint as the average over scores in a

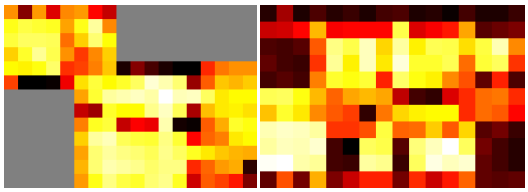


Figure 3: Visualizations of user evaluations of viewpoints. **Left**: House scene, **Right**: Office scene.

House		Office	
Algorithm	**BC**	**Algorithm**	**BC**
Mesh saliency	0.9292	Depth map stability	0.8831
Surface area entropy (exp)	0.9566	Rel. object area entropy (exp)	0.9011
Surface area entropy	0.9759	Rel. object area entropy	0.9187
Curvature entropy	0.9764	Mesh saliency	0.9234
Depth map stability	0.9789	Rel. surface area entropy (exp)	0.9273
Object area entropy	0.9814	Surface area entropy (exp)	0.9317
Curvature entropy (exp)	0.9823	Surface area entropy	0.9425
Rel. object area entropy (exp)	0.9831	Visible area ratio	0.9428
Rel. object area entropy	0.9831	Rel. surface area entropy	0.9455
Object area entropy (exp)	0.9847	Curvature entropy	0.9550
Rel. surface area entropy (exp)	0.9848	Curvature entropy (exp)	0.9577
Object uniqueness	0.9854	Object area entropy	0.9607
Rel. surface area entropy	0.9901	Object area entropy (exp)	0.9644
Visible area ratio	0.9939	Object uniqueness	0.9729

Table 1: Bhattacharyya coefficients (BC) of user evaluations and the algorithm results for both scenes. A higher value corresponds to higher similarity. Results marked with (exp) denote exponentiated versions of logarithmically scaled measures.

0.30m×0.30m regular grid (0.05m increments) centered over the position, including heights from 1.45m to 1.80m in 0.05m increments, to account for small user movements. The user evaluations were compared to the algorithm results by calculating Bhattacharyya coefficients for each measure (see Table 1, other measures produce very similar results). For logarithmic measures, their exponentiated score was also included, as the user evaluation was linear in scale.

4.2. Discussion

In the house, most locations in all rooms got high scores, with the highest scores in the kitchen, where many different objects can be found. The less-detailed patio was rated lower on average, but higher where participants could look into the kitchen or living room. In the office scene, the meeting/seminar rooms and the offices got similarly high scores, while the corridors were rated lower and the empty offices very low.

In the algorithm comparison of the house scene, *visible area ratio* and *relative surface area entropy* show the highest correspondence to user evaluations. However, this is likely due to the fact that the empty outside area—which got very high scores, but would probably have been rated low by humans—was not included in the study. This problem is avoided by the next best algorithms, *object uniqueness* and *object area entropy (exp)*. These also reach the highest correspondence in the office scene, followed by *object area entropy* and *curvature entropy (exp)*. Note that *mesh saliency* seems to produce similar scores as *object uniqueness* on the office scene at first glance. However, its high ratings are found almost exclusively in one area, while most of the scene is rated very low, leading to a bad correspondence to user ratings. In total, *object uniqueness* and *object area entropy (exp)* perform well on both scenes regarding user ratings.

Note that, as this rating scheme weighs all points equally, the correspondence results for a certain algorithm depend on the area occupied by points where it performs well. For example, in the office scene, many points were in empty rooms, making it more important for an algorithm to produce similar ratings as humans there, than it was, e.g., for views from within potted plants. Although we tried to choose scenes that are representative for their class, the recommendations from the study are not necessarily valid for scenes with too differently distributed content. In these cases, conclusions should be drawn directly from the algorithm results and the discussion in section 3.2.

5. Applications

The results of accurate VQE techniques on scenes can be used to support users of IVEs in various ways. Three possible applications are addressed in this section.

Automated Virtual Tours
Knowing the viewpoint quality of all points in a scene is a useful aid for calculating automated virtual tours that give the user an overview of the environment as a replacement for manual exploration. Based on VQE results, efficient virtual tours can be computed that cover a certain percentage of all *interesting* content, instead of, e.g., just trying to see all landmarks once [ETT07]. Approaches for this exist (e.g., [BDP00, AVF04]), but can be improved, for example by optimizing the path itself to not only connect way points efficiently, but to be as informative as possible itself, and by using the most suitable algorithms presented here. Such tours could make exploration tasks, e.g., in automated architectural walkthrough scenarios, much more efficient.

Automatic Travel Speed Selection
In scenes with a varying degree of information density, traveling with a constant speed is often either imprecise or inefficient. However, giving the user manual control over the speed is often undesired as well, as it necessitates the regulation of an additional degree of freedom. This is especially apparent in multi-scale scenarios, where automated travel speed control based on distances to the surrounding scene has been used successfully (e.g., [MMGK09, TR11]). However, these approaches only produce mixed results in narrow (e.g., indoor) environments (where users want to speed up in empty corridors) [TR11] due to the constantly close proximity of floors and walls. Choosing the travel speed based on viewpoint quality instead—letting users go faster when there is actually less to see—can avoid this problem.

Automatic Selection of Representative Views
When a scene has to be presented statically in images, VQE algorithms can be used to generate representative views

(e.g., [SPFG05, FSG09]). In this case, the view direction and field of view should be factored into the viewpoint quality. Note that this is possible with all algorithms described in section 2.1 without changing their structure.

6. Conclusion and Future Work

In this paper, we introduced three new VQE algorithms and compared their performance, together with six existing algorithms, on two different scenes. Furthermore, we conducted a user study to obtain a quantitative evaluation. The results show that the decision of which algorithm performs best depends on the scene and the application. However, our comparison and user evaluation show that *object uniqueness* and *object area entropy* performed consistently well on both tested scenes.

Nevertheless, the results may not necessarily generalize to other *classes* of scenes. Therefore, in future work, we plan to compare these algorithms on more types of scenes, especially containing outdoor areas and other classes of rooms (e.g., factories, churches or lecture halls). Moreover, motivated by the promising results of our new, object-based approach on scenes, we want to further pursue the concept of objects for VQE algorithms. Among others, we plan to improve object descriptors, include object *complexity* into the measure, improve object detection by merging object parts, and try to determine and favor objects that are relevant to the user's current task.

Finally, we will use and evaluate the best-performing algorithms for some of the applications mentioned in section 5, especially automatic travel speed adjustment in scenes with strongly varying information density, and the generation of most informative automatic virtual tours.

References

[Ale12] ALEXANDRE L. A.: 3D Descriptors for Object and Category Recognition: A Comparative Evaluation. In *Workshop on Color-Depth Camera Fusion in Robotics, IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* (2012), vol. 1. 3

[AVF04] ANDÚJAR C., VÁZQUEZ P., FAIRÉN M.: Way-Finder: Guided Tours through Complex Walkthrough Models. In *Computer Graphics Forum* (2004), vol. 23, pp. 499–508. 1, 7

[BDP00] BARRAL P., DORME G., PLEMENOS D.: Visual Understanding of a Scene by Automatic Movement of a Camera. In *Int. Conf. 3IA* (2000). 1, 7

[Bha43] BHATTACHARYYA A.: On a measure of divergence between two statistical populations defined by their probability distributions. *Bull. Calcutta Math. Soc. 35* (1943), 99–109. 3

[CT12] COVER T. M., THOMAS J. A.: *Elements of Information Theory*. John Wiley & Sons, 2012. 2

[DCG10] DUTAGACI H., CHEUNG C. P., GODIL A.: A Benchmark for Best View Selection of 3D Objects. In *Proc. ACM Workshop on 3D Object Retrieval* (2010), pp. 45–50. 1

[DCG12] DUTAGACI H., CHEUNG C. P., GODIL A.: Evaluation of 3D Interest Point Detection Techniques via Human-Generated Ground Truth. *The Visual Computer 28*, 9 (2012), 901–917. 1

[DHKP05] DE HAAN G., KOUTEK M., POST F. H.: IntenSelect: Using Dynamic Object Rating for Assisting 3D Object Selection. In *IPT/EGVE* (2005), pp. 201–209. 1

[ETT07] ELMQVIST N., TUDOREANU M. E., TSIGAS P.: Tour Generation for Exploration of 3D Virtual Environments. In *Proc. ACM Symposium on Virtual Reality Software and Technology* (2007), pp. 207–210. 7

[FSG09] FEIXAS M., SBERT M., GONZÁLEZ F.: A Unified Information-Theoretic Framework for Viewpoint Selection and Mesh Saliency. *ACM Trans. on Applied Perception 6*, 1 (2009). 8

[FWBK15] FREITAG S., WEYERS B., BÖNSCH A., KUHLEN T. W.: Comparison and Evaluation of Viewpoint Quality Estimation Algorithms for Immersive Virtual Environments – Additional Material, 2015. doi:10.18154/2015-04703. 2, 4, 5

[IKN98] ITTI L., KOCH C., NIEBUR E.: A Model of Saliency-Based Visual Attention for Rapid Scene Analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence 20*, 11 (1998), 1254–1259. 2, 3

[JTP06] JAUBERT B., TAMINE K., PLEMENOS D.: Techniques for Off-line Scene Exploration Using a Virtual Camera. In *Int. Conf. 3IA* (2006), vol. 6. 1, 2

[LVJ05] LEE C. H., VARSHNEY A., JACOBS D. W.: Mesh Saliency. In *ACM Trans. on Graphics* (2005), vol. 24, pp. 659–666. 1, 3

[MMGK09] MCCRAE J., MORDATCH I., GLUECK M., KHAN A.: Multiscale 3D Navigation. In *Proc. ACM Symposium on Interactive 3D Graphics and Games* (2009), pp. 7–14. 1, 7

[OFCD01] OSADA R., FUNKHOUSER T., CHAZELLE B., DOBKIN D.: Matching 3D Models with Shape Distributions. In *Int. Conf. on Shape Modeling and Applications* (2001), pp. 154–166. 3

[PKS*03] PAGE D. L., KOSCHAN A., SUKUMAR S. R., ROUI-ABIDI B., ABIDI M. A.: Shape Analysis Algorithm Based on Information Theory. In *Int. Conf. on Image Processing* (2003), vol. 1, pp. 229–232. 1, 2

[PPB*05] POLONSKY O., PATANÉ G., BIASOTTI S., GOTSMAN C., SPAGNUOLO M.: What's in an image? *The Visual Computer 21*, 8-10 (2005), 840–847. 1, 2

[RC11] RUSU R. B., COUSINS S.: 3D is here: Point Cloud Library (PCL). In *IEEE Int. Conf. on Robotics and Automation* (2011), pp. 1–4. 3

[SP05] SOKOLOV D., PLEMENOS D.: Viewpoint Quality and Scene Understanding. In *Int. Conf. on Virtual Reality, Archaeology and Intelligent Cultural Heritage* (2005), pp. 67–73. 1, 2

[SPFG05] SBERT M., PLEMENOS D., FEIXAS M., GONZÁLEZ F.: Viewpoint Quality: Measures and Applications. In *Proc. Eurographics Workshop on Computational Aesthetics in Graphics, Visualization and Imaging* (2005), pp. 185–192. 1, 2, 8

[Tau95] TAUBIN G.: Estimating the Tensor of Curvature of a Surface from a Polyhedral Approximation. In *Fifth Int. Conf. on Computer Vision* (1995), pp. 902–907. 2, 3

[TR11] TRINDADE D. R., RAPOSO A. B.: Improving 3D Navigation in Multiscale Environments using Cubemap-Based Techniques. In *Proc. ACM Symposium on Applied Computing* (2011), pp. 1215–1221. 7

[Váz09] VÁZQUEZ P.-P.: Automatic View Selection through Depth-Based View Stability Analysis. *The Visual Computer 25*, 5-7 (2009), 441–449. 3

[VFSH01] VÁZQUEZ P.-P., FEIXAS M., SBERT M., HEIDRICH W.: Viewpoint Selection using Viewpoint Entropy. In *VMV* (2001), vol. 1, pp. 273–280. 2

[VFSH03] VÁZQUEZ P.-P., FEIXAS M., SBERT M., HEIDRICH W.: Automatic View Selection Using Viewpoint Entropy and its Application to Image-Based Modelling. In *Computer Graphics Forum* (2003), vol. 22, pp. 689–700. 1, 2