Adaptive Web-Based VR Streaming of Multi-LoD 3D Scenes via Author-Provided Relevance Scores

Hendrik Lievens *
Hasselt University - tUL
Expertise Centre for Digital Media

Maarten Wijnants †
Hasselt University - tUL - Flanders Make
Expertise Centre for Digital Media

Mike Vandersanden ‡
Hasselt University - tUL

Peter Quax §
Hasselt University - tUL - Flanders Make
Expertise Centre for Digital Media

Wim Lamotte ¶
Hasselt University - tUL
Expertise Centre for Digital Media

Abstract

The growing storage requirements of 3D virtual scenes, combined with the increased heterogeneity of consumption devices, trigger the need for novel, on-demand streaming techniques of textured meshes. This paper proposes a way to perform *relevance-aware* Adaptive Bit-Rate (ABR) scheduling using MPEG-DASH, tailored to VR consumption in the web browser. Scene authors can annotate the relative importance of scene assets to optimize scheduling decisions. Our approach outperforms the state-of-the-art (measured using the MS-SSIM metric) across different scene complexities and network configurations, and is found to be most beneficial when scene complexity is high and network conditions are relatively poor.

Index Terms: Information systems—Information systems applications—Multimedia information systems—Multimedia streaming

1 Introduction

Three-dimensional scenes allow for six Degrees of Freedom (6 DoF: three axes of translation and three axes of rotation), in addition to a temporal component (cf. video streaming). Consumers have both a position and view-frustum, and are free to interact with the 3D world in an unconstrained and seemingly non-predictive way. As such, existing video-based Adaptive Bit-Rate algorithms (ABR), which typically assume linear consumption and often only consider spatial resolution and network throughput, are not directly translatable to the streaming of 3D scenes. Another subjective consideration for the ABR algorithm deals with bandwidth allocation and resulting perceptual quality. Given a specific bandwidth budget, it might be possible to download either a high fidelity geometry with a low-resolution texture or a high-resolution texture with a low fidelity model.

One of the most recent works in this domain is the DASH-3D approach by Forgione et al. [4]. In DASH-3D, geometry and textures are separated into dedicated MPEG-DASH `AdaptationSets`, so that DASH clients, residing entirely on the client thus eliminating complex server architectures, can compromise between the visual importance of respectively the geometry and texture of a 3D model in terms of bit allocation. DASH-3D is concerned with the network streaming of polygon soups, lacking semantic scene information. Furthermore, it lacks support for multi-resolution geometry.

*e-mail: hendrik.lievens@uhasselt.be
†e-mail: maarten.wijnants@uhasselt.be
‡e-mail: mike.vandersanden@student.uhasselt.be
§e-mail: peter.quax@uhasselt.be
¶e-mail: wim.lamotte@uhasselt.be

To address these two limitations, we propose an improved system that does support geometry with level of detail (LoD) and an accompanying Adaptive Bit-Rate algorithm for textured 3D scenes that understands and exploits the underlying structure of the streamed scene. The premise here is that knowing which objects are more perceptually important than others will aid in deciding the priority with which objects should be scheduled for download. This knowledge can intuitively be thought of as a heat-map, where different parts of the scene are colored according to their relative significance. This heat-map is downloaded alongside the initial scene description, is static, and needs to be provided by the creator of the scene To evaluate our ABR logic, we compare it against Forgione's DASH-3D.

2 System Overview and Implementation

We consider 3D scenes that are comprised of geometry information, i.e. meshes in conjunction with a diffuse texture. To be able to support progressive LoD, Progressively Ordered Primitive (POP) buffers [3] were used. The distinct levels of the POP buffers allow for discrete levels of detail. POP buffers also enable the system to perform instanced rendering, as one POP buffer can appear multiple times within a scene. For textures, three discrete versions are created by spatially scaling the original texture into 2048x2048, 1024x1024, and 512x512 resolutions, each compressed to a JPEG with the highest quality settings. These empirically determined spatial texture resolutions strike a balance between visual fidelity and storage requirements.

The actual structure of the scene is captured in a standards-compliant X3D file [2]. An important difference of the proposed system in comparison with traditional X3D renderers, is that the assets are not downloaded beforehand, but are adaptively requested when needed. The client only needs to reference the scene hierarchy in the X3D file, that contains all the positions, bounding-boxes and other metadata necessary to determine if a given asset is, or will be, in view. This is done by identifying objects within the user's virtual frustum by casting imaginary rays, originating from the user, and returning only those objects that intersect with them, such that only they will be scheduled. Objects have a few important properties that will later aid with their scheduling i.e.:

Area: The total 3D surface area of the object. Determines how big certain objects are within the scene, and is static.

Euclidian distance (between user and object)**:** Objects that are further away will appear smaller and have less overall visual impact.

Utility (adopted from Forgione et al. [4])**:** The **area** of the object divided by the squared **Euclidian distance**. It is used to asses the level of importance of a certain object.

Area ratio: The ratio of the **area** of a single object with respect to the sum of the **area** of all objects in the current scheduling iteration.

Relevance score: An imaginary scalar (unsigned real number) for an object's **area**, directly impacting its **utility**. A value of 100 means that the original **area** of an object will be maintained, while a value of 600 means that the system is tricked into assuming that the object is six times larger than its actual size. A value below 100 will decrease the fictional **area**. These relevance scores are stored in a separate JSON file, the **heat-map**.

All assets (i.e., 3D models, textures, scene X3D file(s), heat-map) are referenced in the DASH manifest. Each asset is stored in a separate `AdaptationSet` and is individually streamable. The `AdaptationSets` are provided with an ID, bandwidth attribute and other relevant data. In the case of geometry information, a custom **area** attribute is added such that the **utility** score of the geometry can be computed without downloading the asset. The geometry and texture `AdaptationSets` encompass multiple `Representations` to enable quality-variant streaming.

The main complexity lies within the client, written in TypeScript and Three.js (supporting WebXR). The server is a stateless file-server. The process starts by retrieving the desired DASH manifest and X3D scene file. The X3D nodes are mapped to DASH `AdaptationSets` by using the same identifier, also referenced in the heat-map file. When this metadata has been retrieved, the client starts scheduling the required assets back-to-back in a desired LoD, depending on the type of ABR logic employed. This process is repeated in the background while the user is already interacting with the scene until all assets have been retrieved in their highest level of detail.

The **DASH-3D ABR logic** was adapted to work specifically with our scene format and geometry representation. It works as follows: all 3D models and textures known to the DASH client are sorted according to their **utility**, in descending order, and the resource with the highest **utility** is always chosen to be scheduled first.

Our proposed **Relevance ABR logic** does not differ all too much from the DASH-3D ABR logic. Besides calculating the **utility** score of the requested resources, the **relevance** scores are also consulted and taken into account to calculate a **scaled utility**. The Relevance ABR logic also calculates the **area ratio** of each object; if that score is below a certain threshold, the non-scaled utility score is used instead of the scaled utility score. The rationale here is that a small area ratio means that for all pending resources, the current object only makes up a small portion and can delay other, perceptually likely more noticeable resources.

3 Experimental Results

We let a virtual camera follow a pre-determined path by interpolating between a number of target positions in the scene. At every frame, a screenshot is taken. Corresponding screenshots from 10 runs of the same path are averaged together and are then concatenated and encoded to a video. This video is compared with a distortion-free reference video, acquired by traversing the same path in a fully downloaded 3D scene, containing only the highest level of detail of objects and textures. For our objective evaluation, we resort to the image-based MS-SSIM index [1]. Calculating the MS-SSIM scores for each frame on the path allows for a one-to-one comparison between different ABR algorithms, i.e. a higher value indicates that the frame at a specific point resembles the ideal result more than a lower value. A comparative study between the DASH-3D ABR logic and the introduced Relevance ABR logic was performed with three specific combinations of network speed and latency: 100 Mbps, 25 Mbps and 5 Mbps, with respective latencies of 10ms, 25ms, 50ms. Additionally, three different scenes have been designed, each with a specific set of features. These scenes and their associated heat-map are shown in Figure 1; the heat-map color coding is as follows: yellow corresponds with the default value (i.e., 100), with green being a higher value and blue being even more important; unimportant objects (i.e., a score lower than 100) are shown in red. Because scheduling gains diminish when more bandwidth is available (i.e., the higher throughput allows more assets to be downloaded in a smaller time frame, thus correcting previous scheduling mistakes), we only report the results for the three evaluated scenes in combination with the slowest network configuration.

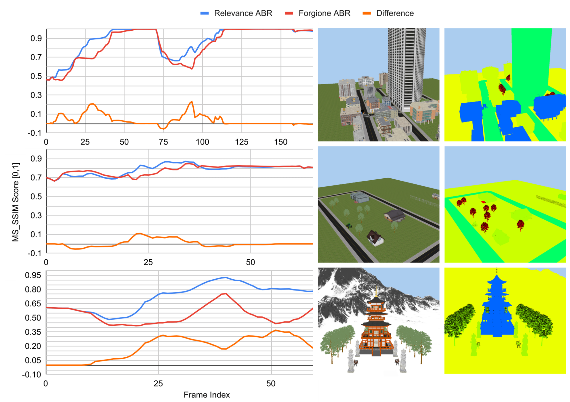

Figure 1: From top to bottom: City, Rural and Temple scene, accompanied by the MS-SSIM scores for the lowest bandwidth setting.

After the start-up phase for the City scene, where the buildings within the starting region are assigned a higher relevance score, new regions of the scene are being discovered, roughly starting at around frame 75, that require scheduling. By exploiting the relevance scores, the Relevance ABR logic knows which assets will contribute the most and can thus make better decisions.

In case of the Rural scene, the DASH-3D ABR logic appears to have a slight edge on the Relevance ABR logic in its startup phase, due to the fact that the DASH-3D ABR logic schedules the grass texture first. After this initial phase, the Relevance ABR logic ignores the less important trees, allowing the hangar to be retrieved earlier.

In the Temple scene, the maximal MS-SSIM score is never reached because the test was terminated before all required assets could be downloaded. We see that the Relevance ABR logic scores up to 35% higher around frame index 50. The lower score attained by the DASH-3D ABR logic is caused by its decision to download the mountain (situated in the background) early on in the asset scheduling process, due to it's huge surface area and thus high **utility** score. Initially, the lowest quality representation is chosen, which is a very coarse approximation and covers more of the sky than the actual mountain. This occlusion leads to a lower score. On the other hand, the Relevance ABR logic decides to schedule the temple, with its distinct red color, first (due to its **relevance** score). Since the temple is also present in the reference frames at these times, the Relevance ABR logic scores a higher MS-SSIM value.

Our evaluation has shown that the proposed scheduling technique holds the potential to improve upon existing ABR techniques, by prioritizing the download of 3D assets that carry more subjective importance, depending on current networking conditions, scene density and lay-out, and the correct usage of relevance scores. While previous research has shown that the MS-SSIM metric is a decent estimator for subjective user perception, it is not guaranteed that this is always the case. User-testing should be the next logical step in assessing how well MS-SSIM scores map to users' QoE.

References

[1] G. Lavoué, et al. On the efficiency of image metrics for evaluating the visual quality of 3d models. In *IEEE Transactions on Visualization and Computer Graphics*, 2016. doi: 10.1109/TVCG.2015.2480079

[2] ISO/IEC 19775-1. Extensible 3D (X3D), 2008.

[3] M. Limper, et al. The pop buffer: Rapid progressive clustering by geometry quantization. In *Computer Graphics Forum*, 2013. doi: 10.1111/cgf.12227

[4] T. Forgione, et al. Dash for 3d networked virtual environment. In *Proceedings of ACM MM*, 2018. doi: 10.1145/3240508.3240701