

TauBench: Dynamic Benchmark for Graphics Rendering

Joel Alanko^a, Markku Mäkitalo^b and Pekka Jääskeläinen^c

Tampere University, Finland

Keywords: Rendering, Graphics File Formats, Virtual Reality, Animation.

Abstract: Many graphics rendering algorithms used in both real time games and virtual reality applications can get performance boosts by reusing previous computations. However, the temporal reuse based algorithms are typically measured using trivial benchmarks with very limited dynamic features. To this end, we present two new benchmarks that stress temporal reuse algorithms: EternalValleyVR and EternalValleyFPS. These datasets represent scenarios that are common contexts for temporal methods: EternalValleyFPS represents a typical interactive multiplayer game scenario with dynamically changing lighting conditions and geometry animations. EternalValleyVR adds rapid camera motion caused by head-mounted displays popular with virtual reality applications. In order to systematically assess the quality of the proposed benchmarks in reuse algorithm stress testing, we identify common input features used in state-of-the-art reuse algorithms and propose metrics that quantify changes in the temporally interesting features. Cameras in the proposed benchmarks rotate on average $18.5\times$ more per frame compared to the popular NVidia ORCA datasets, which results in $51\times$ more pixels introduced each frame. In addition to the camera activity, we compare the number of low confidence pixels. We show that the proposed datasets have $1.6\times$ less pixel reuse opportunities by changes in pixels' world positions, and $3.5\times$ higher direct radiance discard rate.

1 INTRODUCTION

A pleasant and interactive virtual 3D experience requires the display to update a new image in high frequency, but rendering a realistic image takes time. However, the next frame is usually very coherent with the previous one (Yang et al., 2009), even when the rendered content is very dynamic. This coherency can be utilized in order to decrease the computational effort of rendering. These are called *temporal reuse* methods, which means that the previously rendered image is used in some way to accelerate the computation to render a new one.

Often when the performance of methods and processes is compared, *benchmarks* are created and used. Benchmarks contain reproducible test scenarios that are used as an input for algorithms. The results can then be compared with the confidence that the test was performed in a fair setting.

For temporal reuse algorithms, a benchmarking setting would be a dataset that contains 3D data and animations required in the image rendering. It would

be easier to compare algorithm development advancements with standard benchmarks, having access to previously understood and used dynamic datasets. Moreover, such a benchmark would benefit the field of temporal rendering by showing how and where the state-of-the-art algorithms succeed and fail in rendering high-quality animations. It would also serve as a challenge to motivate pushing rendering development forward.

However, there are very few such datasets released in public, and graphics research rarely uses them. There are at least two obvious reasons for this. First, authors create the datasets themselves, own an IP they can use, or buy a set with animations that cannot be released to the public. When such datasets are only present in their research papers, it serves as a potential bias towards the novelties the researchers are proposing, as it is impossible to reproduce the same case. Furthermore, gathering and creating these datasets takes time and effort, and polishing them to release quality would increase it even more (Tamstorf and Pritchett, 2019). This significant time investment to develop datasets tends to be avoided, resulting in the datasets having uninteresting animations and raising the bar to release them. Second, because very few

^a <https://orcid.org/0000-0003-3068-2295>

^b <https://orcid.org/0000-0001-8164-0031>

^c <https://orcid.org/0000-0001-5707-8544>

datasets have been released, in varying file formats, there is no single clear dataset format to select from because there are plenty of standard file formats used across the industry. In summary, the academic research papers use datasets with animations to produce convincing results, but the datasets are rarely publicly available or released to the public.

In this paper, we propose TauBench, which consists of two new dynamic benchmarks that provide a challenge for the temporal reuse methods. These datasets, ETERNALVALLEYVR and ETERNALVALLEYFPS are the first publicly released dynamic benchmarks with a permissive license (CC-BY-NC-SA 4.0) containing interactive rendering use contexts. The animations in the benchmark dataset are more challenging for temporal rendering than any previously published datasets. We evaluate this through various metrics, comparing the fundamental temporal properties that most reuse methods utilize.

Our main contributions in this paper are the following:

- We publish a new dynamic benchmark TauBench with two datasets: ETERNALVALLEYVR: a VR camera benchmark representing a realistic interactive VR use case, and ETERNALVALLEYFPS: a fast-paced camera benchmark that represents the typical interactive first-person application use case. The datasets are available at <https://zenodo.org/record/5729574>.
- We present comparisons on the temporal aspects of the TauBench benchmarks and show that both datasets have a significantly higher motion with the camera, lights, skeletal and rigid body objects in the animation. Comparisons are made by measuring the positional and rotational camera’s activity, the number of new pixels appearing to the camera’s frustum, and the change in the features used by temporal reuse methods.

2 PREVIOUS WORK

2.1 Temporal Coherency

Techniques that utilize frame-to-frame coherency are standard practices across the graphics rendering industry (Yang et al., 2020). However, there have not been temporal complexity comparisons made between different 3D dataset animations to the best of our knowledge, but temporal coherence has been utilized with different input features. We highlight a few families of techniques where dynamic benchmarks can be particularly beneficial, namely anti-aliasing,

upsampling, asynchronous reprojection used in virtual reality applications, solving the path tracing integral with low sample count reconstruction using denoising, and deep learning reconstruction. Temporal coherency methods are discussed more thoroughly in (Scherzer et al., 2012).

Mueller et al. introduced a temporal change metric that detects when colors differ by a perceptually noticeable amount (Mueller et al., 2021). They compare RGB values of tone mapped frames with a set of thresholds, and show with user studies when the pixels differ too much to be under the just-noticeable-difference limit. We use this color change threshold metric in our comparisons.

To detect geometry edges and adapt the denoising filter based on them, McCool proposed comparing adjacent pixels’ normalized color distance, world space position difference, and normal orientation differences (McCool, 1999). Since then, similar denoisers have been utilizing additional feature buffers containing base color, shading normal, world space position, and direct and indirect radiance (Hanika et al., 2011; Li et al., 2012; Sen and Darabi, 2012; Kalantari et al., 2015). The feature comparisons presented in (McCool, 1999) also serve as a basis for some of our comparison metrics.

Temporally caching radiance and computationally taxing irradiance has also seen reuse techniques (Ward and Heckbert, 1992; Krivánek et al., 2005; Tawara et al., 2004), which utilize extra features radiance, irradiance, depth and object identifiers.

Converging the radiance and irradiance to the final image after a short period of time is used in updating irradiance light probes (Gilbert and Stefanov, 2012; Majercik et al., 2019) and dynamic virtual point lights (Keller, 1997; Wald et al., 2003; Hedman et al., 2017; Tatzgern et al., 2020). We use the direct and indirect radiance in our dataset comparison metrics.

2.2 Published Dynamic Datasets

NVidia Open Research Content Archive (ORCA) is a professionally-created 3D assets library from 2017 that has been openly released to the research community (Amazon Lumberyard, 2017; Nicholas Hull and Benty, 2017). The Bistro datasets were created to demonstrate new anti-aliasing and transparency features of the Amazon Lumberyard engine and the Emerald Square dataset to go along with the release of research renderer Falcor (Benty et al., 2020; Amazon.com, Inc., 2021). There is also a dataset by “Beeple” called Zero Day that has a high count of dynamically changing scenery and emissive triangles (Winkelmann, 2019). All the files are in

FBX file format, and they contain camera animations, modern textures, and modern geometry complexity. The datasets run for 60–100 seconds, and their animated cameras have 11–17 key frames described. The ORCA datasets have the most modern geometrical and material representation, with over a million surface faces and physically-based materials, and they feature an animated camera that flies through the dataset. These features make them the current state of the art in dynamic benchmarks. None of these datasets were released for benchmarking purposes in mind, but they are great examples of what kind of datasets are often used by the research community when investigating temporal reuse.

A Benchmark for Animated Ray Tracing (BART) was released in 2001 (Lext et al., 2001). It has three datasets: Kitchen, Robots, and Museum. All of the datasets are described with an infrequently used file format called Animated File Format (AFF), which is an extension of a file format called Neutral File Format (NFF) (Haines, 1987), providing properties to describe animations. The test suite has been released with benchmarking purposes to measure ray tracing performance and has been used in dynamic ray tracing research. Each dataset is designed with a specific stress goal in mind. The Kitchen dataset has significant differences in the density of the details, memory cache performance with hierarchical and rigid body animations, and varying frame-to-frame coherency in the animations. The Robots dataset focuses on the hierarchical animation, distribution of objects in the dataset, and bounding volume overlapping, and the Museum dataset focuses on the efficiency of ray tracing acceleration structure rebuilding. They also propose methods to measure and compare errors when datasets are used with ray tracing algorithms.

The Utah repository collection was released by Wald in 2001 (Wald, 2019). The datasets were released along with two research articles focusing on dynamic ray tracing (Wald et al., 2007; Gribble et al., 2007). The motivation behind setting up the repository with the datasets was that ray tracing was becoming viable for interactive applications. These sets are described with a series of *OBJ* files and use *MTL*s to define the material models, and they do not have camera descriptions released, as they mainly focus on the ray tracing acceleration aspect with these datasets. Wavefront *OBJ* is a human-readable file format to describe 3D geometry and rendering primitives, and the *MTL* file format describes the colors, textures, and reflection maps (Bourke, 2011; Ramey et al., 1995). Describing an animation with a series of *OBJ* files means that the triangles are animated separately, which is commonly also called morph targets or key

shapes (Alexa et al., 2000).

Various other released 3D datasets have temporal aspects. The Moana Island Scene is a complete animation dataset featured in the 2016 Disney film *Moana* (Walt Disney Animation Studios, 2018). The open source 3D animation short film *Sintel* has been used in the creation of the MPI *Sintel* Flow Dataset to be used in motion flow algorithm research (Butler et al., 2012). In addition to *Sintel*, the Blender Foundation provides plenty of openly available datasets in their demo files, displaying the new features for their rasterizer and path tracer (Blender Foundation, 2021). UNC Dynamic Scene Benchmarks have animations of breaking objects and non-rigid object deformations (The GAMMA research group at University of North Carolina, 2018). Similarly, the KAIST Model Benchmarks have animated fracturing objects, cloth simulations, and walking animated characters (Sung-eui, 2014). The downside of these datasets is the lack of temporally challenging scenarios. They either have slowly moving cameras, aged material models, or do not contain moving lights and objects.

3 TEMPORAL RENDERING COMPARISON METRICS

3.1 Camera's Activity

A virtual reality camera can move with six degrees of freedom (DOF): the camera can translate in three directions in XYZ coordinate space, and rotate around three axes with pitch, yaw, and roll. In a typical first-person application controlled with a mouse, the roll rotation is restricted, resulting in five DOF. We calculate the distance d_p the camera travels each frame:

$$d_p(p_i, p_{i-1}) = \| \overrightarrow{p_{i-1}p_i} \|, \quad (1)$$

where p_i is the camera's position on this frame, and p_{i-1} its position in the previous frame.

We also calculate the amount of rotation d_r that happens between frames for each of the three axes with

$$d_r(\theta_i, \theta_{i-1}) = \| \theta_i - \theta_{i-1} \|, \quad (2)$$

where θ_i is the angle in pitch, yaw, or roll rotation for the current frame, and θ_{i-1} the angle for the previous frame.

Moreover, we determine whether the pixels are outside of the frustum with a discard function

$$f_{frustum}(x, y, w, h) = \begin{cases} 1, & \text{if } (x < 0) \parallel (w - 1 < x), \\ 1, & \text{if } (y < 0) \parallel (h - 1 < y), \\ 0, & \text{else,} \end{cases} \quad (3)$$



Figure 1: Rendering of the Sponza scene. From left to right: reference orientation, camera turned 25 degrees from the reference with pitch rotation upwards, yaw rotation to the left, and roll rotation. The checkerboard pattern shows the frustum discard areas that do not have any valid temporal history data. Quick movements, especially in the yaw direction, rapidly invalidate all the available history, whereas roll rotation can invalidate only the corners.

where x, y are the reprojected screen space coordinates and w, h are the width and height of the screen. With $f_{frustum}$, we form a binary discard mask for the frame’s pixels. When the camera has rotation from the previous frame to current in pitch, yaw, or roll rotation, the discarded pixel history is vastly different, as seen in Figure 1.

We apply the frustum discard function to get the discarded percentage $f_{percentage}(w, h)$ of pixels per image by:

$$f_{percentage}(w, h) = \frac{\sum_{i=0}^w \sum_{j=0}^h f_{frustum}(x_i, y_j, w, h)}{wh}, \quad (4)$$

where x_i, y_j are the reprojected coordinates retrieved with indices i, j running through the size of the image’s width w and height h . Finally, we calculate the mean of the discarded pixels for the duration of the animation with $\frac{1}{N} \sum_{i=1}^N f_{percentage_i}(w, h)$, where N is the number of frames in the animation.

3.2 Temporal Feature Comparison

An apparent factor of a temporal challenge is the frame-to-frame change in the feature buffers used by temporal reuse algorithms. When comparing these values for the back reprojected current pixel and the previous frame’s pixel, the distance between the two can determine whether the new pixel is coherent with the previous one. We seek to compare these challenges between datasets. Hence, we form a general metric f_B , similar to the depth-based edge-detection estimator by (McCool, 1999), where we compare the back projected current frame’s feature buffer value B_i to the previous frame’s corresponding value B_{i-1} :

$$f_B(B_i, B_{i-1}, a_B) = \begin{cases} 1, & \text{if } \|\overrightarrow{B_{i-1}B_i}\| > a_B, \\ 0, & \text{else,} \end{cases}$$

where a_B is a confidence threshold value. We use this metric for computing the distance in world space positions, shading normals, direct radiance, and indirect radiance, with the respective thresholds a_{pos} , a_{norm} , a_{dir} , and a_{ind} . Figure 2 illustrates the relevance of the chosen features. The leftmost column

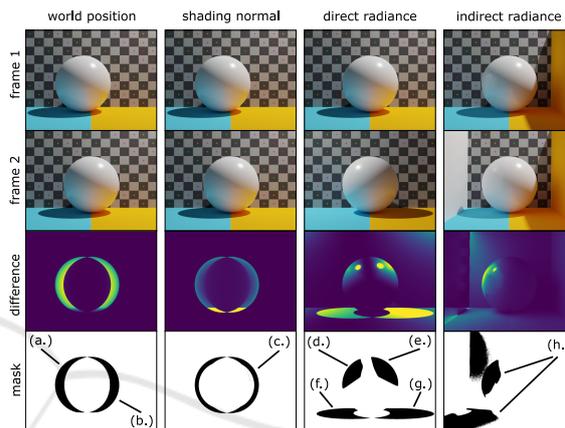


Figure 2: Different masks are composed of the temporal reuse methods’ input features. On the third row, we display the difference between the first and second frames. Comparing the change in pixels’ world positions, we recognize (a.) disocclusions and (b.) occlusions. Disoccluded parts must quickly forget the history buffer, and occluded parts may be reconstructed using back reprojection. When shading normals are compared, we recognize too big of a change in (c.) normal directions between the two frames. The normal angle has changed so much, and we should validate the usability of these pixels. On the third column, a light has moved from right to left, resulting in four different temporally unstable parts: (d.) new direct light, (e.) previously lit, (f.) previously shadowed, and (g.) newly shadowed areas. Finally, on the rightmost column, an appeared wall reflects diffusely (h.) new indirect light on the scene.

shows how occlusions and disocclusions can be recognized by comparing the distance in world positions. Temporal methods try to restart the temporal history for the disoccluded pixels appearing behind the sphere, and reproject the pixels occluded by the sphere with the help of the motion vectors. Mueller et al. presents in (Mueller et al., 2021) that the temporal change in pixel colors can stay unnoticed by human when it changes less than $16/255$ with 8 bit RGB colors. Inspired by their work, we tune our thresholds to $32/255$. This makes sure we compare pixel changes that would most likely be noticed, and to mitigate the effect of each dataset’s geometry being scaled differently. Running the function f_{pos} through the pixels in a frame yields a mask like the one shown in the

leftmost column of Figure 2, containing all the pixels either occluding or disoccluding the geometry. Similarly, columns 2–4 demonstrate the respective masks obtained with the shading normal metric f_{norm} , direct radiance metric f_{dir} , and indirect radiance metric f_{ind} . In particular, the shading history may become less valid, when the reprojected pixels’ shading normal angle changes drastically, or when lights change their position or moving geometry obstructs a shaded point, or when after a few bounces, the light reaches places not previously lit. The confidence thresholds for these metrics are also tuned for each dataset.

4 DYNAMIC BENCHMARKS

4.1 Capturing

Temporally challenging properties are intrinsic aspects of games, so we captured the animation datasets from an open source multiplayer arena shooter game called Cube 2: Sauerbraten (Oortmerssen, 2021). Sauerbraten was selected for the capturing as it was openly available, contained the geometry in common triangle format, and the content was fast paced. We captured two datasets from different areas of the Eternal Valley map, released with a permissive Creative Commons Attribution-NonCommercial-ShareAlike 4.0 Unported License (CC BY-NC-SA 4.0). The map contains the geometry of a sizable outdoor scenery, with the sky directly casting sunlight to the valley, illuminating half of it. Moreover, we updated the scenery with modern GGX materials. The scenery now uses textures with 2K resolution in the base color, normal map, metallic map, and roughness maps. Using the modern open source 3D file format specification glTF 2.0, we encapsulated the datasets to only two singular files: ETERNALVALLEYFPS and ETERNALVALLEYVR. A representative selection of rendered frames of the two datasets are shown in Figure 3.

The first-person camera movement that we capture in ETERNALVALLEYFPS is one of the most common modes in interactive games. It aggregates those essential aspects of interactive scenarios that produce highly temporally changing rendering settings: rapidly changing camera position and irregularly rotating camera orientation. The quick changes around the dataset put a burden on the rendering methods utilizing geometry occlusions and disocclusions, and the camera changes tax the handling of new pixels revealed outside of its frustum.

We used the Unity game engine with the Oculus Quest 2 virtual reality headset to capture HMD cam-

era movement for ETERNALVALLEYVR. The movement is unique, as the head turns and rotates quickly in a way not possible in traditional PC interactive applications. The head is in constant motion and rotates around each axis.

We capture the activity of cameras and geometry animations 60 times per second, for 6 seconds. Comparison datasets are overly lengthy: typical fps selection for real time context is 60, so for example BISTROEXTERIOR that is 100 seconds long, comparison made against ground truth images would require rendering over 6000 frames with high sample count. With 6 seconds we find a balance of understandable and useful content in the animations for comparisons, and rendering time of ground truth images.

Both of the datasets, and rendered videos of them, are available at <https://zenodo.org/record/5729574>.

4.2 Temporal Measurements and Discussion

We render all datasets with Blender’s path tracer Cycles and extract feature buffers, namely world-space positions and normals, direct and indirect radiance, separately. Blender’s Cycles is a path tracing renderer with many supported features, like skeletal animations. We render all of the features with a 1920×1080 resolution, and the pixels are sampled with 1024 paths with a maximum of 12 light bounces. Animations are rendered at 24 fps to guarantee at least some change even in the most stable datasets. With these configurations, we have a reasonable rendering time and a high enough sample count for the indirect buffer to converge enough, mitigating most of the path tracing noise.

We compare our benchmark with modern popular datasets in the ORCA library by NVidia, the BISTROINTERIOR, BISTROEXTERIOR (Amazon Lumberyard, 2017), and EMERALDSQUARE (Nicholas Hull and Benty, 2017). These sets have not been released as temporal rendering benchmarks, but they represent well the datasets often used by the research community, as they have modern geometrical complexity and physically-based material models. In addition, we compare with the TOASTERS dataset from the Utah repository (Wald, 2019), which consists of a vertex morphing setting that is apparent in most of the previously released rendering benchmarks (The GAMMA research group at University of North Carolina, 2018; Sung-eui, 2014). For the comparison we normalized all scenes to human sized scale.

The animation details of the TauBench datasets and the comparison datasets are presented in Table 1.

Table 1: Animation details of the datasets.

	dynamic rigid objects	armatures	static point lights	dynamic point lights
ETERNAL VALLEY FPS	477	32	73	617
ETERNAL VALLEY VR	491	34	63	626
TOASTERS	-	-	-	-
BISTRO INTERIOR	0	0	4	0
BISTRO EXTERIOR	15	0	1	0
EMERALD SQUARE	0	0	2	0

Table 2: Change in camera’s position.

	average	variance	max
ETERNAL VALLEY FPS	1.892	0.826	5.475
ETERNAL VALLEY VR	0.041	0.160	7.163
TOASTERS	0	0	0
BISTRO INTERIOR	0.013	0.000	0.018
BISTRO EXTERIOR	0.055	0.001	0.143
EMERALD SQUARE	0.154	0.004	0.248

In particular, the proposed datasets have significantly more dynamic rigid objects and armatures. ETERNALVALLEY is a large scene, so most of the changes may not be affecting the final render, but the proposed datasets still have much more dynamic features compared to the other sets, as the highest comparison count is on the BISTROEXTERIOR, which has 15 small lamp bulbs swaying slowly by the force of the wind. Moreover, our datasets have over a thousand dynamic point lights appearing, moving and disappearing throughout the animation. In contrast, the only dynamic light sources in the comparison datasets are the animated bulbs in the BISTROEXTERIOR, which have emissive texture on them. However, in our case, the scene is under direct sunlight, making the effect of these bulbs on the final shading negligible.

The amount that the camera translates around the datasets varies significantly, as seen in Table 2. The dataset ETERNALVALLEYFPS moves around the scene the most and has the highest singular per-frame change in position compared to the other datasets. Another noticeable aspect in the movement of the proposed datasets is the continuous small changes in its motion compared to other datasets, which is shown in the more considerable variance. In contrast, the other datasets keep the constant value for a while and then jump abruptly to a new one. This shines some light on the main difference between the proposed dataset and the previous work: our dataset’s animation key frames have been recorded during the game play with high frequency, whereas the previous work has the key frames placed by the animator, letting the camera fly between marked points linearly.

The proposed datasets also have a more significant change in all rotation angles during the camera animation than any other set, as shown in Table 3. Furthermore, compared to the others, the dataset ETERNALVALLEYVR is the only one with considerable roll rotation. This is explained by it being captured with a

Table 3: Change in camera’s rotation per axis, in degrees per frame.

	pitch avg	pitch var	yaw avg	yaw var	roll avg	roll var
ETERNAL VALLEY FPS	1.550	2.839	3.365	8.887	0.000	0.000
ETERNAL VALLEY VR	3.174	10.539	8.222	72.563	1.871	3.469
TOASTERS	0	0	0	0	0	0
BISTRO INTERIOR	0.053	0.004	0.266	0.046	0.002	0.000
BISTRO EXTERIOR	0.014	0.000	0.265	0.051	0.004	0.000
EMERALD SQUARE	0.044	0.002	0.511	0.095	0.000	0.000

Table 4: The percentage of discarded pixels averaged over the length of the animation.

	frustum avg %	color avg %	world position avg %	shading normal avg %	direct radiance avg %	indirect radiance avg %
ETERNAL VALLEY FPS	6.0 %	17.9 %	10.7 %	29.9 %	31.6 %	3.6 %
ETERNAL VALLEY VR	15.3 %	8.3 %	3.1 %	20.3 %	15.1 %	0.8 %
TOASTERS	0.0 %	5.6 %	1.8 %	10.9 %	5.3 %	0.0 %
BISTRO INTERIOR	0.1 %	6.8 %	1.6 %	13.8 %	2.9 %	1.2 %
BISTRO EXTERIOR	0.3 %	6.4 %	2.9 %	20.3 %	7.7 %	0.4 %
EMERALD SQUARE	0.3 %	10.2 %	6.5 %	25.8 %	8.1 %	0.4 %

virtual reality setup, in which the user is constantly swaying their head slightly during the recording. The most active compared dataset EMERALDSQUARE has the highest average in yaw rotation of the previous work, but it is still $16\times$ smaller than the proposed dataset. Moreover, its variance is over $5250\times$ smaller in pitch rotation and $764\times$ smaller in yaw rotation.

In Table 4, we can see that both of the proposed datasets have a higher percentage of pixels that should be discarded due to view frustum changes. The average per-frame frustum discard percentage is 6 – 15% with the ETERNALVALLEY datasets, whereas the highest compared dataset EMERALDSQUARE only has 0.3%. This lines up with the previously recognized change in the camera’s motion. The same trend continues with all of the compared properties, ETERNALVALLEYFPS having the highest rate of low confidence pixels, and ETERNALVALLEYVR the second most. The dataset EMERALDSQUARE does have a reasonably large average percentage with world position and shading normal compared to the other ORCA sets. This is most likely explained due to the amount of high-frequency vegetation the dataset has, as the park in the EMERALDSQUARE is filled with bushes and trees.

The proposed ETERNALVALLEY datasets also show more low confidence pixels due to lighting conditions: both of them have lot of change in direct



Figure 3: Path traced frames from different moments in datasets. On top ETERNALVALLEYFPS, and on the bottom ETERNALVALLEYVR.

lighting condition, and ETERNALVALLEYFPS has the most significant change in the indirect radiance of all the datasets. BISTROINTERIOR and EMERALD-SQUARE do also present change in their direct radiance, but less than the proposed sets.

5 CONCLUSIONS

We have presented TauBench: two new datasets, ETERNALVALLEYVR and ETERNALVALLEYFPS containing an excellent basis to benchmark temporal rendering. The datasets present significantly more temporal complexity than the previously released ORCA datasets.

The proposed datasets contain actual rendering settings captured from a game, which more realistically represent the reuse methods' use of context. When comparing the camera activity with the position and rotational velocity, we show more remarkable change and variation for the animation duration than the compared ORCA datasets. We also showed increased temporal reuse challenge per auxiliary feature buffer, including world position, shading normal, direct, and indirect radiance compared to earlier work. Associated features are often used as input for temporal rendering methods.

The proposed TauBench datasets are more dynamic in many aspects: There are thousands of dynamic rigid body objects, whereas the highest compared dataset contains only 15. The TauBench sets also have 32 skeletal armatures moving around the scene, with 52 animated bones each, whereas the

compared datasets have none. Moreover, there are vastly more static and dynamic lights compared to the previously released rendering datasets. The previous work mainly relies on the sunlight and the environment in the dataset, and the highest count of static lights is in the Bistro Interior with 4 point lights. Both of the proposed datasets surpass this by having over 70 static point lights and 600 animated point lights throughout the animation.

In the future, it would be interesting to extend our dataset comparisons to separate direct and indirect radiance comparisons to diffuse, glossy, transmissive, and volumetric, as now they are all compared in a combined sum. Individually handling the materials would more closely represent how they are handled in typical renderers. We also invite different rendering fields to benchmark their state-of-the-art reuse methods with these two dynamic datasets, with perceived and analytical image quality comparisons, and the methods compute performance. In addition to rasterization and path tracing, the dynamic benchmarks could also be used with real-time virtual point lights, VR, or light field rendering. It would also be interesting to have additional comparisons against other published datasets, like the ray tracing benchmark BART (Lext et al., 2001) and the Bleep's ZERO-DAY dataset in ORCA (Winkelmann, 2019).

ACKNOWLEDGEMENTS

This project has received funding from the ECSEL Joint Undertaking (JU) under Grant Agreement No 783162 (FitOptiVis). The JU receives support from the European Union's Horizon 2020 research and innovation programme and Netherlands, Czech Republic, Finland, Spain, Italy. It was also supported by European Union's Horizon 2020 research and innovation programme under Grant Agreement No 871738 (CP-SoSaware). The project was also supported in part by the Academy of Finland under Grant 325530.

REFERENCES

- Alexa, M., Behr, J., and Müller, W. (2000). The morph node. In *Proceedings of the fifth symposium on Virtual reality modeling language (Web3D-VRML)*.
- Amazon Lumberyard (2017). Amazon Lumberyard Bistro, Open Research Content Archive (ORCA). <https://developer.nvidia.com/orca/amazon-lumberyard-bistro>.
- Amazon.com, Inc. (2021). Demo files. <https://aws.amazon.com/lumberyard/downloads/>.

- Benty, N., Yao, K.-H., Clarberg, P., Chen, L., Kallweit, S., Foley, T., Oakes, M., Lavelle, C., and Wyman, C. (2020). The Falcor rendering framework. <https://github.com/NVIDIAGameWorks/Falcor>.
- Blender Foundation (2021). Demo files. <https://www.blender.org/download/demo-files/>.
- Bourke, P. (2011). Wavefront .obj file format specification. <http://paulbourke.net/dataformats/obj/>.
- Butler, D. J., Wulff, J., Stanley, G. B., and Black, M. J. (2012). A naturalistic open source movie for optical flow evaluation. In *European Conf. on Computer Vision (ECCV)*, Part IV, LNCS 7577. Springer-Verlag.
- Gilabert, M. and Stefanov, N. (2012). Deferred illumination in Far Cry 3. <https://www.gdcvault.com/play/1015326/Deferred-Radiance-Transfer-Volumes-Global>.
- Gribble, C. P., Ize, T., Kensler, A., Wald, I., and Parker, S. G. (2007). A coherent grid traversal approach to visualizing particle-based simulation data. *IEEE Transactions on Visualization and Computer Graphics*, 13(4).
- Haines, E. (1987). Neutral File Format. <http://netghost.narod.ru/gff/vendspec/nff/index.htm>.
- Hanika, J., Dammertz, H., and Lensch, H. (2011). Edge-optimized à-trous wavelets for local contrast enhancement with robust denoising. In *Computer Graphics Forum*, volume 30. Wiley Online Library.
- Hedman, P., Karras, T., and Lehtinen, J. (2017). Sequential Monte Carlo instant radiosity. *IEEE Transactions on Visualization and Computer Graphics*, 23(5).
- Kalantari, N. K., Bako, S., and Sen, P. (2015). A machine learning approach for filtering Monte Carlo noise. *ACM Transactions on Graphics*, 34(4).
- Keller, A. (1997). Instant radiosity. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*.
- Krivánek, J., Gautron, P., Pattanaik, S., and Bouatouch, K. (2005). Radiance caching for efficient global illumination computation. *IEEE Transactions on Visualization and Computer Graphics*, 11(5).
- Lext, J., Assarsson, U., and Moller, T. (2001). A benchmark for animated ray tracing. *IEEE Computer Graphics and Applications*, 21(2).
- Li, T.-M., Wu, Y.-T., and Chuang, Y.-Y. (2012). SURE-based optimization for adaptive sampling and reconstruction. *ACM Transactions on Graphics*, 31(6).
- Majercik, Z., Guertin, J.-P., Nowrouzezahrai, D., and McGuire, M. (2019). Dynamic diffuse global illumination with ray-traced irradiance fields. *Journal of Computer Graphics Techniques*, 8(2).
- McCool, M. D. (1999). Anisotropic diffusion for Monte Carlo noise reduction. *ACM Transactions on Graphics*, 18(2).
- Mueller, J. H., Neff, T., Voglreiter, P., Steinberger, M., and Schmalstieg, D. (2021). Temporally adaptive shading reuse for real-time rendering and virtual reality. *ACM Transactions on Graphics*, 40(2).
- Nicholas Hull, K. A. and Benty, N. (2017). NVIDIA Emerald Square, Open Research Content Archive (ORCA). <https://developer.nvidia.com/orca/nvidia-emerald-square>.
- Oortmessen, W. v. (2021). Cube 2: Sauerbraten. <http://sauerbraten.org/>.
- Ramey, D., Rose, L., and Tyerman, L. (1995). MTL material format (Lightwave, OBJ). <http://paulbourke.net/dataformats/mtl/>.
- Scherzer, D., Yang, L., Mattausch, O., Nehab, D., Sander, P. V., Wimmer, M., and Eisemann, E. (2012). Temporal coherence methods in real-time rendering. In *Computer Graphics Forum*, volume 31. Wiley Online Library.
- Sen, P. and Darabi, S. (2012). On filtering the noise from the random parameters in Monte Carlo rendering. *ACM Transactions on Graphics*, 31(3).
- Sung-eui, Y. (2014). Korea Advanced Institute of Science and Technology (KAIST) model benchmarks. <https://sglab.kaist.ac.kr/models/>.
- Tamstorf, R. and Pritchett, H. (2019). The challenges of releasing the Moana Island Scene. In *Eurographics Symposium on Rendering - DL-only and Industry Track*. The Eurographics Association.
- Tatzgern, W., Mayr, B., Kerbl, B., and Steinberger, M. (2020). Stochastic substitute trees for real-time global illumination. In *Symposium on Interactive 3D Graphics and Games*.
- Tawara, T., Myszkowski, K., and Seidel, H.-P. (2004). Exploiting temporal coherence in final gathering for dynamic scenes. In *Proceedings Computer Graphics International, 2004*. IEEE.
- The GAMMA research group at University of North Carolina (2018). UNC dynamic scene benchmarks. <https://gamma.cs.unc.edu/DYNAMICB/>.
- Wald, I. (2019). Utah animation repository. <https://www.sci.utah.edu/~wald/animrep/index.html>.
- Wald, I., Benthin, C., and Slusallek, P. (2003). Interactive global illumination in complex and highly occluded environments. In *Rendering Techniques*.
- Wald, I., Boulos, S., and Shirley, P. (2007). Ray tracing deformable scenes using dynamic bounding volume hierarchies. *ACM Transactions on Graphics*, 26(1).
- Walt Disney Animation Studios (2018). Moana Island Scene (v1.1). <https://www.disneyanimation.com/data-sets/?drawer=/resources/moana-island-scene/>.
- Ward, G. J. and Heckbert, P. S. (1992). Irradiance gradients. Technical report, Lawrence Berkeley Lab., CA (United States); Ecole Polytechnique Federale.
- Winkelmann, M. (2019). Zero-Day, Open Research Content Archive (ORCA). <https://developer.nvidia.com/orca/beeple-zero-day>.
- Yang, L., Liu, S., and Salvi, M. (2020). A survey of temporal antialiasing techniques. In *Computer Graphics Forum*, volume 39. Wiley Online Library.
- Yang, L., Nehab, D., Sander, P., Sitthi-amorn, P., Lawrence, J., and Hoppe, H. (2009). Amortized supersampling. *ACM Transactions on Graphics*, 28(5).