A Distributed, Decoupled System for Losslessly Streaming Dynamic Light Probes to Thin Clients
Supplemental Material

Michael Stengel
NVIDIA
mstengel@nvidia.com

Zander Majercik
NVIDIA
amajercik@nvidia.com

Benjamin Boudaoud
NVIDIA
bboudaoud@nvidia.com

Morgan McGuire
NVIDIA
mcguire@nvidia.com

1 PROBE ATLAS ARRANGEMENT
As each texel represents directional information it is obvious that adjacent probes contain angular redundancy. We have tested different texel arrangements for the probe atlas. The original DDGI paper uses 10x10 pixels for the core probe color information (Figure 3, left). We rearrange texels from four probes in a 2x2 block (Figure 3, center) and from eight probes in a 4x4 block (Figure 3, right) so that texels of neighboring probes but with similar directions during probe tracing are adjacent in the probe atlas. This way adjacent pixels often contain similar values which support algorithms such as Run Length Encoding (RLE) or Delta Encoding. However, our analysis reveals that the lossless HEVC video encoder could exploit redundancy best in the original layout where individual probes are arranged as individual blocks. This result holds for lossless as well as for lossy compression (see compressed sizes in Figure 3). HEVC compression works on texel blocks which can be quite similar for multiple probes across the probe atlas with respect to different compression metrics. The video encoder therefore has a better chance in exploiting redundancies across all probes in a volume than from just angular probe redundancy of neighboring probes.

2 NETWORK THROUGHPUT
We show the consumed bandwidth over time using our different bandwidth reduction methods (Figure 16-12).

- **Uncompressed**: direct uncompressed texture transfer, Section 3.2
- **Encoded**: full texture compression, Section 3.3
- **Culling**: update only probes in client view, send full atlas, Section 3.4
- **Culling, Packing, No caching**: update only probes in client view, reduce atlas to active and updates probes, no cache for previously updated probes, Section 3.4
- **Packing and Caching**: as before but previously transmitted probe indices are cached for probe placement in atlas to increase temporal stability of probe textures, Section 3.4

3 SELECTIVE PROBE TRANSMISSION
We discuss our recommended strategy for selective probe transmission in Section 3.4 in the paper. Our method computes all probes shading points visible in the client’s camera view frustum and then augments that set with an estimate of all probes shading points visible from the camera’s current position. This is computed on the server by rasterizing the client’s primary view and then tracing 1024 rays uniformly distributed on a sphere centered at the camera position. The raster pass and ray tracing pass combined yield a set of world space positions. All probes that shade these positions are then marked for transmission.

To further evaluate our selection mechanism, note that only probes that have changed texel values since they were last transmitted could possibly convey useful information to the client. Therefore, the number of changed probes is an upper bound on the number of probes that should ever be sent. From the other direction, all probes that have new data and shade points in the client’s primary view frustum must be sent to correctly render the current view on the client. The number of these probes is the lower bound on the number of probes sent. As shown in Figure 2, our selection mechanisms achieves the image quality of the higher bound while transmitting a number of probes closer to the lower bound. Further, the number of rays used to gather potentially visible points around the camera can be adjusted depending on the constraints of image quality vs. bandwidth.

| Render | Tomb Buddha | Greek Villa | Night Street |
|--------|-------------|-------------|--------------|
| Mean (ms) | 3.49 | 6.39 | 3.57 |
| Std. Dev. (ms) | 0.34 | 0.46 | 0.34 |
| Min (ms) | 2.36 | 5.50 | 3.15 |
| Max (ms) | 6.48 | 7.38 | 5.54 |

Table 1: Client render times. Timings evaluated on gaming GPU (NVIDIA RTX 2080 Ti).

![Figure 1: Thin Client: NVIDIA Xavier NX. The portable Tegra GPU on the NVIDIA Xavier NX developer board provides two low-power hardware units for accelerated HEVC video decoding. Timings for decoding are given in Table 3 in the paper.](image-url)
Figure 2: Selective Probe Transmission. Our Greek Villa scene with 4096 probes. Probe selection is done relative to a client reference view known to the server (top left). Sending only probes within the client primary view gives large outdated regions (top right, outdated regions rendered in black). In this case, 1261 probes are transmitted. All regions are up to date when all active probes are sent (bottom left). In this case, 3299 probes are transmitted. Finally, the same image quality is achieved with our potentially visible probe set estimation (bottom right). In this case, though the image is the same, only 1710 probes are transmitted.

Figure 3: Probe Atlas Arrangement Analysis. Probe atlas sizes are given for Greek Villa scene averaged for 278 frames: (left) 1x1 probe, original DDGI arrangement: 224 kilobytes, (center) 2x2 probes interleaved: 238 kilobytes, (right) 4x4 probes interleaved: 250 kilobytes.
Figure 4: Selective Probe Reduction Performance. Average percentage of probes updated (over time) with unchanged probes removed. The most aggressive culling transmits probes within the primary view only (top), but suffers from artifacts (see Figure 3). Our recommended method transmits more probes by augmenting the primary view set with a potentially visible set of probes from the camera’s perspective (middle). This approach gives the same image as doing no selection but sends significantly fewer probes than transmission without probe selection (bottom).
| Streaming Data | Average Latency (Std.Dev.) | Min/Max Latency | Average Size (Std.Dev.) | Min/Max Size |
|----------------|----------------------------|-----------------|-------------------------|--------------|
| Irradiance     | 37.01 ms (20.361)          | 10 ms / 196 ms  | 10.52 kB (4.06)         | 3.02 kB / 33.941 kB |
| Visibility     | 39.03 ms (20.339)          | 8 ms / 255 ms   | 0.692 kB (4.951)        | 0.09 kB / 97.680 kB |

Figure 5: Detail time series plot for scene **Greek Villa with view culling**. During recording the client moves around in the scene. The lighting in the scene permanently changes due to a fast day-night cycle. Probe visibility is only updated when the user can see the orrery in the scene (around times 0:18 and 0:40) reducing the amount of demanding transfers. Streaming settings: 30 Hz irradiance update rate, 10 Hz visibility update rate, selective probe updates for enabled, delta-encoded index buffer probe index buffer with caching, client view culling enabled.
Figure 6: Detail time series plot for scene **Greek Villa without view culling**. The scene is the same as in the previous plot with the difference that in this plot all changing probes are transferred to the client. The degree of lighting change for the day-night cycle can easily be observed in the cycle of the resulting encoded texture size. Probe visibility is updated whenever the orrery changes position (time 0:20 to 1:15). Streaming settings: 30 Hz irradiance update rate, 10 Hz visibility update rate, selective probe updates for enabled, delta-encoded index buffer probe index buffer with caching, client view culling disabled.
Figure 7: Method overview showing consumed network bandwidth for scene Greek Villa. The overview shows that our probe reduction and compression method reduces the required network traffic by at least one order of magnitude.

Figure 8: Method overview (excluding uncompressed transfer) showing consumed network bandwidth for scene Greek Villa. During recording the client moves on a fixed animation path through the scene. The probe volume update rate has a strong but sub-linear influence on the consumed network bandwidth and can be seen as a ‘quality knob’ controlling the dynamic quality of global illumination (GI) for the client. Furthermore, our probe selection and caching schemes significantly reduce the amount of transferred data. Probe selection alone without caching can actually increase the amount of transferred data as encoding efficiency is decreased. Probe selection in combination with index caching however optimizes for efficient temporal encoding of only changed probes resulting in a significant reduction of bandwidth compared to full volume encoding. The culling method can be tuned for even more aggressive probe reduction if the client field of view is limited further (Fig. 2, not shown here) but can result in perceivable artifacts as the probe update is delayed after quick turns of client.
Figure 9: Detail time series plot for scene Night Street with view culling. During recording the client moves around in the city scene. The lighting in the scene permanently changes locally for moving cars on the street and for a rotating emergency light. Probe visibility is only updated when the user can see the moving cars which happens frequently for some probes during the time of recording. Although the scene contains 8000 probes the transmitted data the probe selection and encoding is surprisingly efficient as at a point in time only a limited number of probes contribute to the view of the client. Streaming settings: 30 Hz irradiance update rate, 10 Hz visibility update rate, selective probe updates for enabled, delta-encoded index buffer probe index buffer with caching, client view culling enabled.
Figure 10: Detail time series plot for scene Night Street without view culling. The scene is the same as in the previous plot with the difference that in this plot all changing probes are transferred to the client. The amount of streamed data is slightly higher than with client view culling enabled. However, our probe selection still significantly and consistently reduces the amount of streamed data with respect to the full probe volume containing 8000 probes. Streaming settings: 30 Hz irradiance update rate, 10 Hz visibility update rate, selective probe updates for enabled, delta-encoded index buffer probe index buffer with caching, client view culling disabled.
Figure 11: Method overview showing consumed network bandwidth for scene Night Street. The overview shows that our probe reduction and compression method reduce the required bandwidth between one and two orders of magnitude.

Figure 12: Method overview (excluding uncompressed transfer) showing consumed network bandwidth for scene Night Street. During recording the client moves on a fixed animation path through the scene. The consumed bandwidth is highest when the number of probes is high (0:07 to 0:12). Probe selection in combination with index caching optimizes for efficient temporal encoding of only changed probes resulting in a significant reduction of bandwidth compared to full volume encoding. Probe selection without caching can actually increase the amount of transferred data as encoding efficiency is decreased. Client view culling in combination with index caching helps to furthermore reduce the number of updated probes and maintain encoding efficiency. As a result, culling and probe selection with index caching enable a significant reduction of bandwidth compared to full volume encoding.
Streaming Data | Average Latency (Std.Dev.) | Min/Max Latency | Average Size (Std.Dev.) | Min/Max Size
--- | --- | --- | --- | ---
Irradiance | 26.33 ms (18.295) | 15 ms / 174 ms | 1.64 kB (5.697) | 0.03 kB / 44.62 kB
Visibility | 30.63 ms (25.925) | 12 ms, Max 204 ms | 12.55 kB (56.536) | 0.04 kB / 419.89 kB

**Figure 13: Detail time series plot for scene Tomb Buddha with view culling.** In this scene the client view includes almost all probes in the scene. Both lighting and visibility in the scene permanently change rapidly when the roof of the building caves in (time 0:18 to 0:35). This is the most challenging scenario for the streaming approach as all probe data must be transmitted resulting in a spike of consumed bandwidth and transfer latency. However, the perceived latency is still well under the perception threshold for diffuse GI latency (< 500 ms) [Crassin et al., CloudLight, 2015]. It is apparent that the costs for changing visibility of all probes in the volume dominate the costs for updating irradiance. Streaming settings: 30 Hz irradiance update rate, 30 Hz visibility update rate, selective probe updates for enabled, delta-encoded index buffer probe index buffer with caching, client view culling enabled.
| Streaming Data | Average Latency (Std.Dev.) | Min/Max Latency | Average Size (Std.Dev.) | Min/Max Size  |
|---------------|---------------------------|-----------------|-------------------------|--------------|
| Irradiance    | 26.21 ms (23.267)         | 14 ms / 180 ms  | 4.36 kB (9.829)         | 0.03 kB / 55.97 kB |
| Visibility    | 30.07 ms (32.731)         | 13 ms / 238 ms  | 26.41 kB (90.008)       | 0.04 kB / 601.44 kB |

**Figure 14: Detail time series plot for scene Tomb Buddha without view culling.** Please see the explanation of the previous plot with the difference that in this plot all changing probes are transferred to the client. As in the previous plot the costs for changing visibility of all probes in the volume dominate the costs for updating irradiance. Streaming settings: 30 Hz irradiance update rate, 30 Hz visibility update rate, selective probe updates for enabled, delta-encoded index buffer probe index buffer with caching, client view culling disabled.
Figure 15: Method overview showing consumed network bandwidth for scene Tomb Buddha. Our probe reduction and compression method reduce the required network traffic by one order of magnitude for this worst-case scenario scene.

Figure 16: Method overview (excluding uncompressed transfer) showing consumed network bandwidth for scene Tomb Buddha. During recording the client moves on a fixed animation path through the scene. The consumed bandwidth is highest when the roof of the building in the scene is caving in (0:03 to 0:06) resulting in a change in visibility and irradiance for almost all probes. Probe selection in combination with index caching optimizes for efficient temporal encoding of only changed probes resulting in a significant reduction of bandwidth compared to full volume encoding. Probe selection without caching can actually increase the amount of transferred data as encoding efficiency is decreased. Client view culling in combination with index caching helps to furthermore reduce the number of updated probes and maintain encoding efficiency.