1. Alternative cache entries

Of course, there are other alternative representations, than our cone representation, that could be used to approximate $L_i$ in a cache. Arguably, one of the most straight-forward approaches would be a constant approximation of $L_i$ for each light source, by e.g. extending the hard shadow binary cache [WTS*23]: For each area light source, we could store one floating point number that represents its partial visibility which we could progressively update using ray tracing. For shading, we could approximate $L_i$ using the stored visibility and concentrate all radiance to the center direction of the light source. Such a simple approach not only lacks detailed directional information during shading, but also suffers for non-uniform light sources, as shown in Figure 1 top. Note how our cone representation not only better captures the light intensity, but also captures which parts of the area light source are visible from which surface points, i.e., that the area under the chair receives more green light than blue.

Another commonly used representation for capturing radiance are spherical harmonics (SH) [Mil06]. SH can not only be used to capture $L_o$, but could also be used for $L_i$. In theory, an SH basis could faithfully capture either and could thus serve as a perfect cache. However, to represent high frequencies, a large number of bands would be necessary, which would not only lead to very high memory requirements, but also computing and updating all those bands would require a high amount of compute. Thus, we present experiments with using three SH bands only, which results in 27 values per cache entry (compared to our 18). As shown in Figure 1 bottom, SH for $L_o$ is not capable of capturing view-dependent effects well. Using SH for $L_i$ can capture some view dependent effects, but these heavily depend on the sampling and thus introduce significant amounts of noise.

For a quantitative evaluation of the different representations, see Table 1. To reach a somewhat fair comparison, we tested on our completely static version of Sponza only and let all approaches run until convergence. In this way, we evaluate the ceiling of the approaches, ignoring that cones converge with the lowest number of samples and require significantly fewer computations compared to SH. SHs cannot capture high frequencies of $L_o$; caching $L_i$ (as all other 3 methods) clearly increases quality.

2. Cache Bias Comparison

Adding a bias to the cache allocation essentially reduces the cache resolution and forces spatial reuse of the same cache data. While a higher bias may significantly increase performance, it also may
reduce quality similar to a shadow map resolution that is too small. We plot the relation between quality and rendering time in Figure 3, again for a static converged scene. Clearly, a bias of up to 1.5 hardly changes quality. As such, our test setup using a bias of 1.0 is still very conservative and we could have reduced rendering times by another 30% with little impact on quality. Example renderings for the different bias values are shown in Figure 2. Even for this difficult scene parts, a bias of 1.5 is hardly distinguishable from 0.0. A bias of 2.0 leads to visible artifacts around the edges of objects and a higher bias clearly shows artifacts for hard shadows and at boundaries.

3. Speed and Size of lights

Edge cases are evaluated in two setups. The first setup 4 includes an emissive sphere moving over 100 units in 10 seconds. Next to its path there are obstacles that cast shadows. We speed up the simulation in order to look how results differ with changing speeds.

One can see that for very high speeds noise starts occurring and a slight lag shows. This lag can of course be combated by using more samples during cone adaption.

To get a feeling how the size of light sources impacts results we scale a light from one square units to a hundred square units. This can be seen in figure 6. For larger sizes noise starts to occur, but we still achieve good results.

4. Additional Multiviewer Timings

To further show effectiveness we show two setups with 60 percent overlap in figure 8. One for our Bistro Exterior (based on [Lum17]) setup and one for our Sponza setup (based on [Int]).
Figure 7: Light Scaling from 1m$^2$ to 100m$^2$ as seen in 6.

Figure 8: Scaling for 1 to 16 viewers in Bistro Exterior and Sponza for 60 percent overlap demonstrating our superior scaling in comparison to screen-space approaches. Note that our approach also achieves superior quality.

References


