LightCluster - Clustering Lights to Accelerate Shadow Computation

Daniel Wiesenhütter*, Andreas Klein†, Alfred Nischwitz‡
Munich University of Applied Sciences

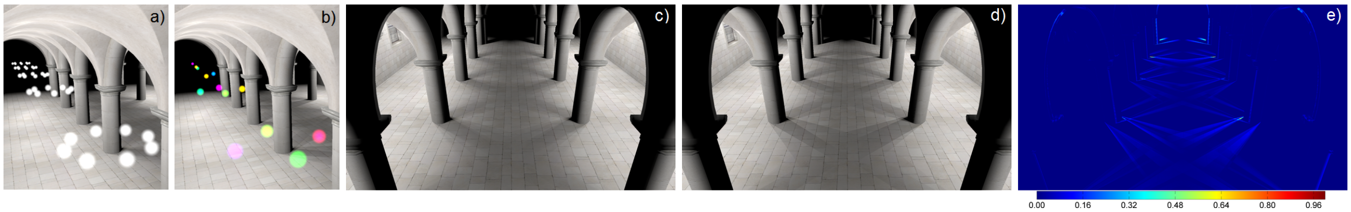

Figure 1: *Comparison between a reference solution and LightCluster for the Dabrovic Sponza scene at a resolution of 1920x1080. For a given distribution of 80 point light sources in both arcades of the atrium (a), only 26 clusters are selected with a minimum distance metric (b). Instead of rendering hard shadows for every point light source like in the reference solution (c) at 97.8 ms, soft shadows are computed for every selected cluster (d) at 43.1 ms with Percentage Closer Soft Shadows. In (e) the difference between the reference solution (c) and the solution of LightCluster (d) is displayed. Notice that dark blue color indicates a low error and a red color a high error.*

1 Introduction

Shadows are an important part of a visualization and give the viewer supplemental details about the appearance of objects. In real-time rendering, shadow mapping [Williams 1978] is a popular approach to compute shadows. However, a shadow map must be computed for each light and thus, the memory and the computation time increases with the number of lights.

We present an approach, called LightCluster, to automatically select representative light sources and accelerate the computation of direct shadows for scenes with many lights. We carefully select light sources as cluster centers and cluster the remaining lights using a minimum distance metric [Wolfowitz 1957]. We represent each cluster by an area light source and use Percentage Closer Soft Shadows (PCSS) [Fernando 2005] to render soft shadows for each cluster. Figure 1 shows an overview.

The idea of [Dong et al. 2009] is closely related to our work. In contrast to Clustered Visibility we focus our work on high frequency shadows for direct lighting. Furthermore, our approach uses an existing light source as a cluster center and clusters the remaining lights with a minimum distance metric.

In our implementation, we use omnidirectional point lights. However, the approach can be adapted for other light types.

2 Our Approach

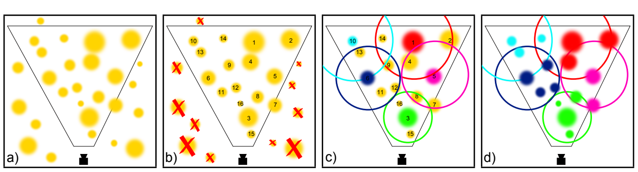

Figure 2: *Clustering in a 2D example scene: (a) point light distribution, (b) frustum culling of point lights, (c) selecting clusters with a minimum distance metric, (d) assigning lights to nearest clusters.*

In order to reduce the error in shadows, we perform a two pass clustering with different metrics in each pass. We first select point lights within the viewing frustum as cluster centers by using the light range and a minimum-distance metric, which is scaled by the camera distance. This allows us to generate smaller clusters and thus, more shadow maps, near the camera position. In the second clustering pass, the remaining point lights are assigned to the nearest cluster centers. Therefore, the error in shadows due to the decreased amount of shadow maps can be reduced. Figure 2 shows the clustering.

Then, we render a cube shadow map for each cluster and interpret the cluster as a disc-shaped area light source. The radius of the area light source is given by the minimum distance of the cluster. We use this radius to scale the filter window of PCSS. The visibility factor is then stored in a texture for each cluster. This allows us to calculate the shadows iteratively and reduces the texture memory from a cube shadow map to a screen sized texture per cluster. In this way, we sample the cube shadow maps only once for each cluster and avoid additional PCSS sampling for each light source during shading. After the shadow computation, we use the set of visibility textures for shading. We shade the scene with each point light source and modulate the resulting color with the visibility value stored in the texture for the point light's cluster.

By adjusting the clustering parameters, a well defined ratio between quality and performance can be chosen. Due to the approximation of the shadows from many lights with a soft shadow from a representative light source, errors in the shadow can be reduced. As the clustering is fast, it can be performed in each frame and only adds a small overhead. If light sources can be merged to clusters, LightCluster can be used to increase the rendering performance while maintaining an acceptable shadow quality in many cases.

References

DONG, Z., GROSCH, T., RITSCHEL, T., KAUTZ, J., AND SEIDEL, H.-P. 2009. Real-time indirect illumination with clustered visibility. In *Proc. of the VMV*, 187 – 196.

FERNANDO, R. 2005. Percentage-closer soft shadows. In *ACM SIGGRAPH 2005 Sketches*, ACM, New York, NY, USA, SIGGRAPH '05.

WILLIAMS, L. 1978. Casting curved shadows on curved surfaces. *SIGGRAPH Comput. Graph. 12*, 3 (Aug.), 270–274.

WOLFOWITZ, J. 1957. The minimum distance method. *Annals of Mathematical Statistics 28*, 1, 75–88.

*e-mail:daniel@wiesenhuetter.me
†e-mail:andreas.klein@hm.edu
‡e-mail:nischwitz@cs.hm.edu

SIGGRAPH 2013, July 21 – 25, 2013, Anaheim, California.

ACM 978-1-4503-2261-4/13/07