Remote and scalable interactive high-fidelity graphics using asynchronous computation

Keith Bugeja[1,2]

[1] University of Malta
[2] University of Warwick

Abstract. Current computing devices span a large and varied range of computational power. Interactive high-fidelity graphics is still unachievable on many of the devices widely available to the public, such as desktops and laptops without high-end dedicated graphics cards, tablets and mobile phones. In this paper we present a scalable solution for interactive high-fidelity graphics with global illumination in the cloud. Specifically, we introduce a novel method for the asynchronous remote computation of indirect lighting that is both scalable and efficient. A lightweight client implementation merges the remotely computed indirect contribution with locally computed direct lighting for a full global illumination solution. The approach proposed in this paper applies instant radiosity methods to a precomputed point cloud representation of the scene; an equivalent structure on the client side is updated on demand, and used to reconstruct the indirect contribution. This method can be deployed on platforms of varying computational power, from tablets to high-end desktops and video game consoles. Furthermore, the same dynamic GI solution computed on the cloud can be used concurrently with multiple clients sharing a virtual environment with minimal overheads.

1 Introduction

Modelling of global illumination increases the level of realism and immersion in virtual environments [8]. While a large number of methods for computing graphics of higher fidelity have been developed, they typically trade off quality for performance and are incapable of running on all but machines with the highest specifications. Cloud computing has enabled the use of low-performance devices for tasks beyond their computational capabilities. Complex tasks are assimilated into cloud services, allowing applications running on these devices to request and receive the results in a fraction of the time it would take the local device to compute. In terms of visualisation and rendering, this model has recently been exploited by providers such as OnLive to provide interactive streaming services for games [5]. A thin client connects to a data-centre in the cloud, where the service provider hosts and runs the actual game, and receives its audiovisual output stream. User input, such as directional controls and button presses, are transmitted by the client to the server, fed to the game and in response, the game output is sent back to the client in the form of a compressed video stream. The bulk of the computation is carried out at the provider's data-centre, allowing a wide range of devices to consume the service, making the computational capacity of the client device largely irrelevant. Although effective in providing the same experience to a plethora of devices with varying capabilities, this paradigm is highly susceptible to network latency and bandwidth constraints. High definition and ultra high definition (UHD) streams, especially at higher frame rates, transfer significant amounts of data (see Table 1), and may exclude some network configurations due to bandwidth limitations or the introduction of undesired lag in programs that require low response times. In these settings, each client connects to an application that performs the rendering in isolation. This one-to-one approach precludes the possibility of rendering algorithms that amortise computation complexity over a number of concurrent clients, such as is the potential with multi-user environments. As opposed to rendering entirely in the cloud, Crassin et al. use an approach similar to what we propose, where the rendering pipeline is only partially offloaded from the client [2]. In particular, they introduce a distributed rendering pipeline which computes the indirect lighting contribution in the cloud, amortising the computations across multiple clients in a multi-user environment. Three lighting algorithms were proposed, each with different bandwidth and reconstruction costs [3][7][4][6]. Two of these algorithms, the path-traced irradiance maps and real-time photon mapping, are asynchronous in nature, decoupling client updates from the cloud computation and the network performance. Irradiance maps yield low bandwidth requirements, and reconstruction costs are also cheap, but the difficulty in acquiring UV-parameterisation for moderately complex scenes doesn't always make them a viable option due to the laborious nature of the parameterisation [2]. Photon tracing doesn't require any parameterisations but has substantially larger bandwidth requirements, close to an order of magnitude more than the requirements of streaming cloud gaming platforms. Moreover, the indirect lighting reconstruction at the client poses prohibitive computational costs for some low to mid-range devices. The third algorithm, which adopts a synchronous approach, uses cone-traced sparse voxel global illumination, and although client updates at 30 Hz can be sustained for 5 clients, this soon drops to 12 Hz as soon as the number of clients is increased to 24. The system, CloudLight, supports vertical scaling by the addition of more GPUs to a server node, but it is unclear as to how the system scales horizontally.

Service	**Resolution**	**Bandwidth**
Netflix	720p	4.0 Mbps
Netflix	1080p	5.0 Mbps
Hulu Plus	720p	2.0 Mbps
Hulu Plus	1080p	3.2 Mbps
OnLive	720p	5.0 Mbps
Playstation Now!	720p	5.0 Mbps

Table 1. Bandwidth requirements for various video-on-demand services.

This work [1] proposes an asynchronous remote computation technique that provides an efficient method for computing indirect lighting. The proposed method has minimal bandwidth and client-side computation requirements, and can achieve client-side updates at 60 Hz or more at HD and UHD resolutions. The global illumination solution is split into two components, direct and indirect lighting. The indirect lighting, which is the most computationally expensive, is decoupled from the rest of the rendering and carried out remotely. This service is provided to clients of multi-user environments, thus amortising the cost of computation over all connected consumers of the service. Computation results are stored in an efficient object space representation that does not require the large bandwidths of the solutions above. Furthermore, transfers being in object space are resolution agnostic; increasing client resolution does not increase bandwidth requirements, as is the case with streaming solutions. The lightweight client reconstruction makes this method suitable for any device that supports basic rendering functionality, while also scaling well to many clients connected to the same service, achieving a significant overall boost in performance.

Fig. 1. Test scenes running at over 60 Hz at full HD resolution (1920 × 1080) using our method for the asynchronous computation of indirect illumination in the cloud.

References

1. K. Bugeja, K. Debattista, S. Spina, and A. Chalmers. Remote and scalable interactive high-fidelity graphics using asynchronous computation. In *Submitted to The 36th Annual Conference of the European Association for Computer Graphics*. EG, 2015.
2. C. Crassin, D. Luebke, M. Mara, M. McGuire, B. Oster, P. Shirley, P.-P. Sloan, and C. Wyman. Cloudlight: A system for amortizing indirect lighting in real-time rendering. 2013.
3. C. Crassin, F. Neyret, M. Sainz, S. Green, and E. Eisemann. Interactive indirect illumination using voxel cone tracing. In *Computer Graphics Forum*, volume 30, pages 1921–1930. Wiley Online Library, 2011.
4. H. W. Jensen. *Realistic image synthesis using photon mapping*. AK Peters, Ltd., 2001.
5. M. Manzano, J. A. Hernández, M. Uruеña, and E. Calle. An empirical study of cloud gaming. In *Proceedings of the 11th Annual Workshop on Network and Systems Support for Games*, page 17. IEEE Press, 2012.
6. M. Mara, D. Luebke, and M. McGuire. Toward practical real-time photon mapping: efficient gpu density estimation. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pages 71–78. ACM, 2013.
7. J. Mitchell, G. McTaggart, and C. Green. Shading in valve's source engine. In *ACM SIGGRAPH 2006 Courses*, pages 129–142. ACM, 2006.
8. K. Myszkowski, T. Tawara, H. Akamine, and H.-P. Seidel. Perception-guided global illumination solution for animation rendering. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 221–230. ACM, 2001.