Collaborative rendering over peer-to-peer networks

Keith Bugeja[1]

Department of Computer Science, University of Malta
keith.bugeja@um.edu.mt

Abstract. Physically-based high-fidelity rendering pervades areas like engineering, architecture, archaeology and defence, amongs others [3][7]. The computationally intensive algorithms required for such visualisation benefit greatly from added computational resources when exploiting parallelism. In scenarios where multiple users roam around the same virtual scene, and possibly interact with one another, complex visualisation of phenomena like global illumination are traditionally computed and duplicated at each and every client, or centralised and computed at a single very powerful server. In this paper, we introduce the concept of collaborative high-fidelity rendering over peer-to-peer networks, which aims to reduce redundant computation via collaboration in an environment where client machines are volatile and may join or leave the network at any time.

1 Introduction

High-fidelity rendering makes use of physically-based quantities to compute light transport through a virtual scene, simulating the interactions between light and surfaces in order to compute energy levels in the scene. The equilibrium of light energy can be formulated mathematically via an integral equation known as the Rendering Equation [8], which has inspired a large variety of rendering algorithms and provides a definition of radiance at any point in the scene:

$$L_o(p, \omega_o) \quad = \quad L_e(p, \omega_o) \quad + \int_{\Omega} f(p, \omega_o, \omega_i) \;\; L_i(p, \omega_i) \;\; |\cos\theta_i| \;\; \mathrm{d}\omega_i \qquad (1)$$

where p is a point in the scene, L_e is the emitted light, ω_i and ω_o are the incoming and outgoing directions of light, f is the bidirectional reflectance distribution function (BRDF) [10] [9], giving the proportion of light reflected from ω_i to ω_o at p, and $|\cos\theta_i|$, ($n_p \cdot \omega_i$, n_p being the normal at p), is the foreshortening factor.

The main challenge in high-fidelity rendering is solving the rendering equation efficiently for any given scene. A number of approaches exist that can be broadly categorised into two methods: finite element (e.g., radiosity [6]) and point-based sampling (e.g., distributed ray tracing [4]). Finite element methods have some limitations which make them less than ideal for the goals of interactive rendering.

The general approach discretises surfaces into a finite set of patches, introducing difficulties when modelling arbitrary geometries. Moreover, solutions that go beyond the radiosity approach and model arbitrary reflections, present enormous problems due to memory consumption and computation times [11]. Most modern methods, such as Path Tracing [8] and Instant Global Illumination [12] use point-based sampling. Monte Carlo techniques, based on ray tracing, are employed for their applicability to arbitrary geometries and reflection behaviours [11]. Another advantage of Monte Carlo techniques is stability; they have the property that error bound is always $\mathcal{O}(n^{-\frac{1}{2}})$ regardless of dimensionality. Point-based sampling methods lend themselves more to parallelisation than finite element methods.

A number of methods have been presented for solving the Rendering Equation within a distributed context. Some solutions use grid computing, desktop grids in particular, to provide high-fidelity rendering to a single client [1]. *Rendering as a Service* uses cloud computing to provide interactive rendering to mutliple clients [2]. In both these works, potential imbalances in load that may occur due to resource volatility are handled by controlling and redistributing tasks to suit resource changes.

In many areas requiring interactive visualisation and collaboration, such as military and architectural simulations or online multiplayer videogames, participants interact with one another in a shared virtual world such as a battlefield, a building walkthrough or even a football pitch. Most of these applications are precluded from using high-fidelity physically-based rendering due to the large computational costs associated with such visualisation. [2] attempted to provide interactive physically-correct rendering by offloading expensive computations to the cloud. The system isolates resources allocated to a single client from other resources in the system, forgoing any communication or collaboration, notwithstanding the fact that sharing the same scene, isolated resources are carrying out duplicate computations.

In this work, we propose a method for exploiting the computation overlap of multiple clients visualising and freely interacting in a shared virtual scene. Furthermore, we assume that the number of constituent clients is not known beforehand; clients are volatile and may join or leave the network at any time. In particular, we put forward a system where computations can be encapsulated into uniquely marked transactions that can contribute towards a distributed shared state. The propagation of transactions neither makes use of an overlay network, nor requires the peer network be fully connected, but instead employs an epidemic paradigm, where peers gossip to exchange these transactions. We envisage the content of transactions to be arbitrary, although currently, a case study is being carried out with irradiance samples generated using [13]. During each exchange, peers also share details on their neighbours, which are then used to update and prune their respective peer neighbourhood directories.

The irradiance cache [13] is an acceleration data structure for caching diffuse irradiance samples from indirect lighting, applied to the distributed ray-tracing algorithm. The algorithm exploits spatial coherence by interpolating these irradiance samples for a given neighbourhood to estimate irradiance in specific regions, thus dramatically reducing render time. Searches within the irradiance cache are carried out to locate valid samples that can be used for irradiance interpolation; in order to speed up these searches, an octree is incrementally built: the diffuse irradiance value is stored and the tree topology updated.

We are looking at extending this work to cater for ordered transactions. For example, in static scenes, the update order of the irradiance cache is irrelevant. However, in dynamic scenes, knowledge of whether a transaction happened before another or not is required in ensuring consistency in the generated output. As such, we are looking at ordering each transaction using arrays of logical clocks [5] to generate a partial ordering of events and detect causality violations. Moreover, given the large number of participant, we are also investigating the application of some form of importance sampling for pruning the vector clock data structure to reduce bandwidth usage.

References

1. Aggarwal, V., Debattista, K., Bashford-Rogers, T., Dubla, P., Chalmers, A.: High-fidelity interactive rendering on desktop grids. Computer Graphics and Applications, pre print (2010)
2. Bugeja, K., Debattista, K., Spina, S., Chalmers, A.: Rendering as a service : Scalable high-fidelity rendering in the cloud. Future Generation Computer Systems (under review) (2013)
3. Chalmers, A., Debattista, K.: Level of realism for serious games. In: Mendez, G.R., Liarokapis, F., Freitas, S.D. (eds.) VS-GAMES (2009)
4. Cook, R.L., Porter, T., Carpenter, L.: Distributed ray tracing. In: ACM SIGGRAPH Computer Graphics. vol. 18, pp. 137–145. ACM (1984)
5. Fidge, C.J.: Timestamps in message-passing systems that preserve the partial ordering. Proceedings of the 11th Australian Computer Science Conference (1988)
6. Goral, C.M., Torrance, K.E., Greenberg, D.P., Battaile, B.: Modeling the interaction of light between diffuse surfaces. In: Computer graphics and interactive techniques. pp. 213–222. SIGGRAPH '84, ACM, New York, NY, USA (1984)
7. Happa, J., Bashford-Rogers, T., Wilkie, A., Artusi, A., Debattista, K., Chalmers, A.: Cultural heritage predictive rendering. Computer Graphics Forum (2012)
8. Kajiya, J.: The rendering equation. In: Computer graphics and interactive techniques (1986)
9. Matusik, W., Pfister, H., Brand, M., McMillan, L.: A data-driven reflectance model. In: ACM SIGGRAPH 2003 Papers. pp. 759–769. SIGGRAPH '03, ACM, New York, NY, USA (2003)
10. Nicodemus, F.E.: Directional reflectance and emissivity of an opaque surface. Appl. Opt. 4(7), 767–773 (Jul 1965)
11. Stamminger, M.: Finite element methods for global illumination computations. Herbert Utz Verlag (1999)
12. Wald, I., Kollig, T., Benthin, C., Keller, A., Slusallek, P.: Interactive global illumination using fast ray tracing. In: Eurographics workshop on Rendering (2002)
13. Ward, G.J., Rubinstein, F.M., Clear, R.D.: A ray tracing solution for diffuse interreflection. In: SIGGRAPH '88 (1988)