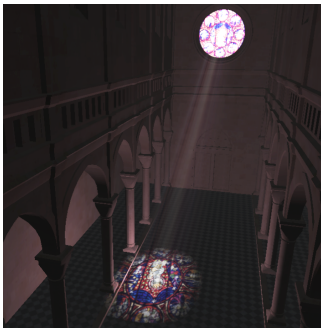
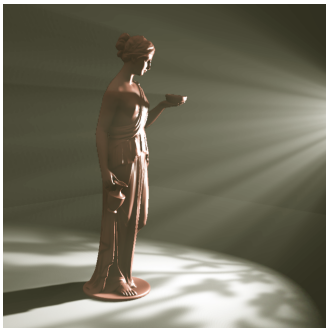


Volumetric Shadow Mapping

Pascal Gautron
Thomson Corporate Research
pascal.gautron@thomson.net

Jean-Eudes Marvie
Thomson Corporate Research
jean-eudes.marvie@thomson.net

Guillaume François
The Moving Picture Company
guillaume-f@moving-picture.com



(a) Hebemissin - 200 steps - 28 fps (b) Dark Alley - 200 steps - 21 fps (c) Sibenik - 200 steps - 50 fps (d) Eddie - 150 steps
Figure 1: The light volume is sampled in shadow map space to compute single scattering, accounting for occlusions and shadows. The computation is carried out using graphics hardware for real-time performance (a, b, c), or RenderMan for production-quality rendering (d).

Classical rendering methods usually consider that light travels in vacuum, hence overlooking its interactions with its medium of transmission: the air. However, interactions between light and particles in suspension in the air generate phenomena such as smoke, fog, dust... Existing methods for simulating such interactions often rely on the native fog attenuation of graphics hardware, ignoring the occlusion effects shown in Figure 1. Mitchell [2005] tackles this problem by shading series of semi-transparent volume slices using a shadow map. While providing visually pleasant results, the method is not physically accurate and is prone to artifacts in given viewing directions due to undersampling.

Our approach builds upon two unrelated rendering techniques: the well-known shadow mapping algorithm [Williams 1978] and sub-surface texture mapping [Francois et al. 2008] for real-time scattering simulation in multilayered materials defined by 2D textures. We simulate single scattering by evenly sampling the 2D light space to reduce aliasing artifacts due to undersampling of the shadow map. The design of volumetric lighting commonly features gobo textures to aesthetically control the light shaft effects. The ray marching in the 2D light space uniformly samples such textures, hence preserving the gobo details which are unlikely to be preserved when sampling the camera ray in world space.

Volumetric Shadow Mapping

Our rendering method, based on shadow mapping (Figure 2), is divided into two passes:

We first render the scene from the camera C into a RGBA buffer in which the 3 first channels encode the radiance received by the visible objects according to both the shadow map and the average extinction of the participating medium. The alpha channel stores the distance from the camera to the closest visible object.

In the second pass we render a screen-aligned quadrilateral bounding the screen region covered by the light cones as seen from C . The fragment shader first intersects the cones with the viewing ray to determine the length of the light path through the cones. The possible presence of opaque objects within the cones is taken into account using the distances computed in the first pass (Figure 2(a)). This path is then projected into the 2D space of the shadow map and sampled using a user-defined number of sampling points (Figure 2(b)). For each point, we determine the light source visibility using the shadow map, and compute the contribution of that point to the total radiance outgoing towards the camera.

We also introduce a *scattered ambient lighting* term to enforce the visual coherency of the lighting throughout the scene.

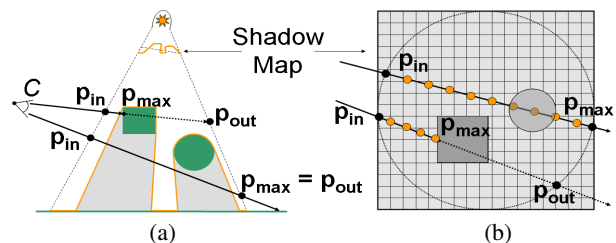


Figure 2: Determination of the traversal distance (a) and ray marching in shadow map space (b).

Results

Our algorithm offers real-time performance on a 3.6GHz Xeon with an nVidia Geforce GTX280. The scenes of Figures 1(a, b, c) are rendered at a resolution of 1280×720 . Production renderers also benefit from our method as it provides a user-defined, robust control of the sampling quality of the light shafts and gobos (Figure 1(d)).

Volumetric Shadow Mapping is a unified method for fast computation of light shafts in participating media, oriented towards the preservation of the lighting features. The sole tuning of the number of ray marching steps yields rendering qualities ranging from real-time to production.

Future work will particularly consider an extension to multiple scattering simulation, as well as an accurate simulation of volumetric penumbras due to translucent objects using deep shadow mapping.

References

- FRANCOIS, G., PATTANAIK, S., BOUATOUCH, K., AND BRETON, G. 2008. Subsurface texture mapping. *IEEE Computer Graphics & Applications* 28, 34–42.
- MITCHELL, J. L. 2005. *ShaderX3: Light Shaft Rendering*. Charles River Media, 573–588.
- WILLIAMS, L. 1978. Casting curved shadows on curved surfaces. In *Proceedings of SIGGRAPH*, 270–274.