Optimized Mobile Rendering Techniques Based on Local Cubemaps

Roberto Lopez Mendez* and Sylwester Bala*
ARM Ltd.

Figure 1: *Shadows, refraction and reflections based on local cubemap in the Ice Cave demo.*

Abstract

Local cubemaps (LC) were introduced for the first time more than ten years ago for rendering reflections [Bjorke 2004]. Nevertheless it is only in recent years that major game engines have incorporated this technique. In this paper we introduce a generalized concept of LC and present two new LC applications for rendering shadows and refractions. We show that limitations associated with the static nature of LC can be overcome by combining this technique with other well-known runtime techniques for reflections and shadows.

Rendering techniques based on LC allow high quality shadows, reflections and refractions to be rendered very efficiently which makes them ideally suited to mobile devices where runtime resources must be carefully balanced [Ice Cave Demo 2015].

Keywords: local cubemap, shadows, reflections, refractions

Concepts: • **Computer graphics ~ Rendering;** *Rasterization;*

1 The concept of local cubemaps

Let's assume we have a local environment delimited by an arbitrary boundary and we have rendered the surrounding environment into a cubemap. We want to find the vector we need to use to retrieve the star from the cubemap texture. To simplify the problem of finding the intersection point P we introduce the bounding box of the scene as a proxy geometry. We build a new vector from the position the cubemap was baked to the intersection point and we use this new "local corrected vector" to fetch the texture from the cubemap.

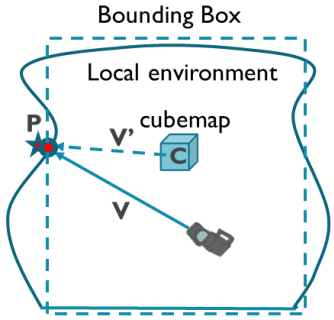

Figure 2: *The concept of local cubemap.*

The lesson here is that we need to apply the local correction for every vector we use to retrieve the data we store in the LC.

2 Soft shadows based on local cubemaps

The concept of this technique is to render the transparency of the local environment boundaries to the alpha channel of a static cubemap off-line. Then at runtime in the shader we use the fragment-to-light vector to fetch the texel from the cubemap and determine if the fragment is lit or shadowed [Bala and Lopez Mendez 2005].

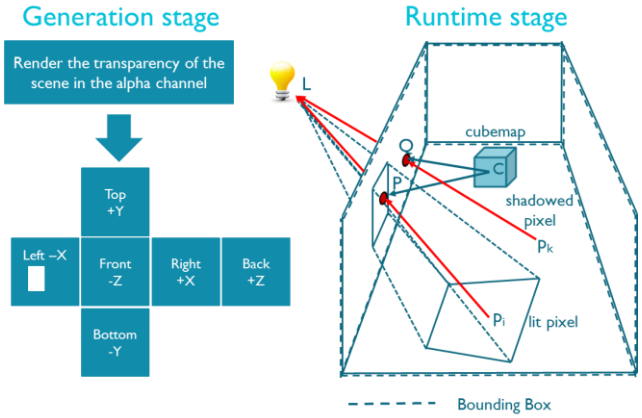

Figure 3: *Soft shadows based on local cubemaps.*

*e-mail:Roberto.LopezMendez@arm.com.
*e-mail:Sylwester.Bala@arm.com.

SIGGRAPH 2016 Posters, July 24-28, 2016, Anaheim, CA.
ACM 978-1-4503-4371-8/16/07.
http://dx.doi.org/10.1145/2945078.2945113

As we are dealing with a LC, the local correction has to be applied to the fragment-to-light vector before the fetching operation.

To render soft shadows we make use of mipmaps. We calculate a coefficient which is proportional to the distance from the pixel to the intersection point. The coefficient is then used to select a mipmap level so the more distant the pixel is from the intersection point the more blurred it gets.

Figure 4: *Soft shadows based on local cubemaps [Chess Room Shadows Demo 2016].*

3 Refraction based on local cubemaps

Refraction is an important effect to consider when striving for extra realism when rendering semi-transparent geometry. Refraction is the change in direction of a light wave due to a change in the transmission medium. When using cubemaps to implement refraction in a local environment, we need to apply the local correction to produce the correct results.

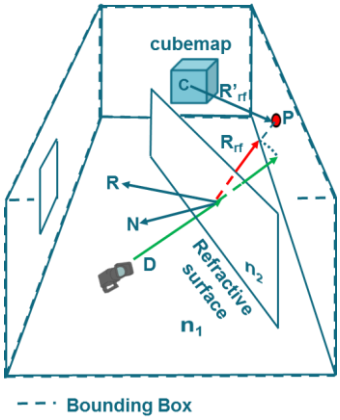

Figure 5: *The local correction to the refraction vector.*

According to the definition of LC provided in the section 1, to retrieve correctly the texture in the direction of the refraction vector R_{rf} we need to apply the local correction to it. After determining the direction of the refracted vector R_{rf}, we need to find where it intersects the bounding box of the local scene. The next step is to build a new vector R'_{rf} from the position where the cubemap was generated to the intersection point P and use this final vector to fetch the texel from the cubemap to render what is behind the refractive object. We get a physically based refraction because the direction of the refraction vector is calculated according to Snell's Law. Moreover, there is a built-in function we can use in our shader to find the refraction vector R strictly according to this law [Lopez Mendez 2015].

3 Solving limitations of local cubemaps

The new shadow technique introduced in section 2 enables rendering shadows from the static geometry of the boundaries of the scene. This technique works nicely in open plan spaces with no other geometry, especially in the center where the cubemap will likely be generated. Additionally, this technique won't produce shadows from dynamic geometry due to the static nature of the cubemap.

Nevertheless, the above mentioned limitations can be easily solved if we combine this technique with other well-known shadow rendering techniques, for example, shadow mapping. In this case all objects, and especially dynamic ones, are removed during the process of baking the cubemap. At runtime the shadows on the boundaries of the scene and on any object will be rendered using the local cubemap technique and the shadows produced by those objects will be rendered using the shadow mapping technique.

The same principle applies to solving the limitations of local cubemaps when rendering reflections. For example, reflections based on local cubemap can be combined with planar reflections rendered at runtime using the mirrored camera technique. The Ice Cave demo shows how shadows and reflections based on LC are effectively combined with other runtime rendering techniques.

4 Performance and quality

The static nature of the LC does have a positive impact in that it allows for faster and higher quality rendering. For example, shadows based on LC are 1.3 - 1.5 times faster than shadow mapping. The fact that we use the same texture every frame guarantees high quality shadows with no pixel instabilities which are present with other shadow rendering techniques.

Finally, as there are only read operations involved when using static LC the bandwidth use is halved. This feature is especially important in mobile devices where bandwidth is a limited resource. The conclusion here is that when possible, use rendering techniques based on LC. When combining with other techniques they allow us to achieve higher quality at very low cost.

References

BALA, S., AND LOPEZ MENDEZ, R. 2016. Efficient Soft Shadows Based on Static Local Cubemap. GPU Pro 7, Chapter IV Mobile Devices, p.175.

BJORK, KEVIN. 2004. Image Based Lighting. GPU Gems, Chapter 19, p. 307.

CHESS ROOM SHADOWS DEMO. 2016. https://www.youtube.com/watch?v=VR4x__JApQ0

ICE CAVE DEMO FXS. 2015. https://www.youtube.com/watch?v=mb98QOlZ8ZE

LOPEZ MENDEZ, R. Refraction Based on Local Cubemap. ARM Connected Community. 2015.