Angular Extent Filtering with Edge Fixup for Seamless Cubemap Filtering

John R. Isidoro **sketches_0268** Jason L. Mitchell
ATI Research ATI Research

Synopsis

We present a new efficient method for cubemap filtering which alleviates the edge seam artifacts that occur with standard cubemap filtering methods. This approach works with current cubemapping hardware since it does not require rarely supported texture border hardware used in other approaches [1].

Despite the fact that cube maps are defined on the spherical domain, standard cubemap filtering techniques perform filtering independently on each cube face. The main problem with this approach is that no information is propagated across edges, thus creating undesirable discontinuities along the cube face edges. A limitation of nearly all cubemapping hardware which makes the seam problem substantially worse is the fact that the bilinear texel filtering is not able to fetch across cube faces thus producing a hard seam artifact. The seam problem also causes aliasing artifacts. These two compounding problems limit the usefulness of cubemapping.

Our filtering approach alleviates these problems using two techniques. The first, angular extent filtering, defines each tap's filter kernel using an angular extent around the center tap as opposed to a fixed per-face pixel-based extent. The advantages of angular extent filtering are that the filtering kernels used for all taps have a constant solid angle as well as have the ability to pull texels from multiple faces. In addition to this, the filtering takes into account the solid angle subtended by each tap in the filter. Also, by varying the angular extent of the filter used to generate the base miplevel, pre-convolved reflections for a variety of materials can be precomputed.

The second technique, edge seam fixup, uses a per miplevel seam averaging and smoothing algorithm in order to effectively hide the hard seam artifacts due to limitations of current cubemapping hardware. This enables even extremely low-resolution cubemap miplevels (even 2x2 and 4x4) to be directly used for tarnished metal shaders, matte surface shaders, and even for diffuse lighting.

Efficient Angular Extent Filtering

In order to efficiently process all the texels within an angular domain around a center tap vector, we use a few different lookup tables to expedite the computation. The key idea is to process each texel within the circular filter extent exactly once in a reasonably quick way. Processing each tap only once is particularly important for HDR imagery, where a single bright pixel value can dominate a region.

To filter cubemaps in this way, we first generate a normalizer and solid angle lookup cubemap the same size as the source cubemap. This is used to quickly lookup the associated direction vector and solid angle for any given texel in the cubemap. Then per output cube-map sample, conservative bounding box regions for the angular kernel extent are computed for each of the cube faces. Within each bounding box region the filter weights are computed per-tap via a dot product between the normalizer cube map lookup for that texel. The dot product is used as a fast test to see whether or not the given tap is within the angular extent of the kernel.

To implement filters using a falloff that is a function of the angle between the current tap vector and the center tap vector, the result of the dot product can used to index into a lookup table to compute the tap weight. We have implemented a variety of different filters using this method such as an angular version of a Gaussian filter, a cosine filter and a cone filter.

Edge Seam Fixup Methods for Cubemap Filtering

The idea behind edge seam fixup is to average cubemap edge texels across edges, and obscure the effect of the averaging by adjusting intensity values within a 1 to 15 texel fixup region around the edges. Since the texel colors on either side of the edge are the same, this eliminates the hard seam artifacts resulting from the inability of the hardware to filter across cubemap edges. There are a few methods we have experimented with in order to improve the results in the fixup region.

Pull Fixup methods adjust the texel intensity values in the edge fixup region by a fraction of the amount the edge pixels were adjusted by during the average. This fraction either falls off cubically to zero as the distance from the edge increases. These methods tend to maintain high frequency information within the fixup region and work best when the cubemaps are highly detailed.

Smoothed Fixup methods perform a weighted average of the texel intensity values in the edge fixup region with the averaged edge values. The weight either falls off cubically to zero as the distance from the edge increases. The smoothed fixup methods are best for diffuse or smoothly varying cubemaps as they tend to remove high frequency detail in the edge regions. They also do a good job of smoothing over edge discontinuities in the case when the source images used to construct the cubemap have different intensities or exposure levels.

Per-Pixel Selective Blurring

Now that all levels of the mipchain are directly useable, the per-pixel miplevel selection ability of modern pixel shader hardware can be used to implement per-pixel roughness mapping. In this case the cubemap is used as an environment map. In the pixel shader, the maximum of the current pixels cubemap miplevel, and the miplevel determined by the roughness map is used to select the appropriate miplevel from the environment map to appropriately blur the reflections.

The leftmost two images show the 2x2 miplevel without (first) and with (second) angular extent filtering and edge fixup The rightmost two images show the 4x4 miplevel without (third) and with (fourth) angular extent filtering and edge fixup. Notice how our novel filtering method removes the seam artifacts and makes even these extremely low resolution cubemaps directly useable.

References

1) ASHIKHMIN, M. and ABHIJEET, G. 2002. *Simple Blurry Reflections with Environment Maps*. In Journal of Graphics Tools, 7(4): 3-8.