Joseph Shumway CSCE 181
830003458 Spring 2022

Hardware Adaptive High-Order Interpolation
Daqi Lin, University of Utah
https://www.youtube.com/watch?v=eGfX1iWzkh0

Lin proposed in his presentation a new formulation of high order interpolation in graphics and other discrete data. For example, Bilinear and Bicubic interpolation in 2d textures. In 3d models, this would look like linear or phong(quadratic) deformation for interpolating smoothed normal in a polygonal model. In volume rendering, this would look like Trilinear or Tricubic interpolation. The problem with these interpolations is that they have high computational cost and low hardware support. The computations are expensive and could be made better. Some GPUs now have bicubic and tricubic support but they still use software for their interpolation and this slows things down. There is still no hardware support for simplexes (triangles, tetrahedra, etc).

His solution is to refactor high order interpolation as standard linear interpolation plus linear interpolations of high order terms. This lets them use already existing hardware and GPUs. They can also support 2D/3D/simplexes fully. Their results are very well within the range of standard interpolation. Where bilinear is 1x computation cost, their method is 1.6x, and standard Bicubic is 5x the computation cost. The resulting image is still quite high in quality with much lower cost.

In a standard bicubic filtering algorithm, you need to first take in point values from the texture L1 cache, then compute the terms, then apply weights. After that you need the Accumulation Weights to calculate the scale, which is then passed to the Accumulator and finally outputted. In their Adaptive Bicubic Texture Filtering algorithm, they skip the operation sections that can easily be interpolated linearly instead of bicubically. This is especially true for images that are composed of smoothly and linearly interpolated values for example a fade to black section. The bicubic and linear functions both result in very similar values, so it is unnecessary to to use the more expensive bicubic function.

For 3D Texture filtering, it works similarly. They can skip the bicubic function if the values will be about the same as a linear texel interpolation function. As far as simplexes, they modify the existing bilinear interpolation unit to support barycentric interpolation. In addition to these, High order interpolation on unstructured triangular meshes can also be accelerated.

The results are quite nice. For the Arcade scene, with no TAA (Temporal antialiasing), you can see jagged/pixelated edges on the chair as expected. With TAA Bilinear, the edges are smoothed decently but there is a lot of excessive blurring done that makes the chair look fuzzy in the background, especially when the camera is in motion. With TAA Bicubic, the edge blurriness is significantly reduced and the chair looks much more separate from its surrounding. Using their Adaptive bicubic function, there is no visible difference in quality, but the computational time is quite reduced. Overall, their algorithm is an effective means to interpolate without high overhead.