Textures

January 16, 2022

A texture is a two dimensional buffer (it can be 1D or 3D too) with extra property. The GPU have some built in ships to make some useful computations. For example, with a 1D texture object you can ask this value:

```
float value = my_texture[1.3f];
```

In this situation, the GPU can return my_texture[1] if filtering property is set to "closest" or the average value between my_texture[1] and my_texture[2] if the filtering property is set to "linear". Asking a continuous value into a buffer is not possible.
In this example, we define a 2D memory for 16x16 pixels.

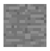

Figure 1: Small image at scale one

After that we want to show thus pixel in a large window. A up-scaling is necessary. To do so, each pixel of the window will read the buffer and choose a pixel to print. On the first image, the strategy is to choose the image pixel the closest to the position of the window pixel in continuous space. On the second image, the strategy is to average the image pixel to avoid some aliasing.

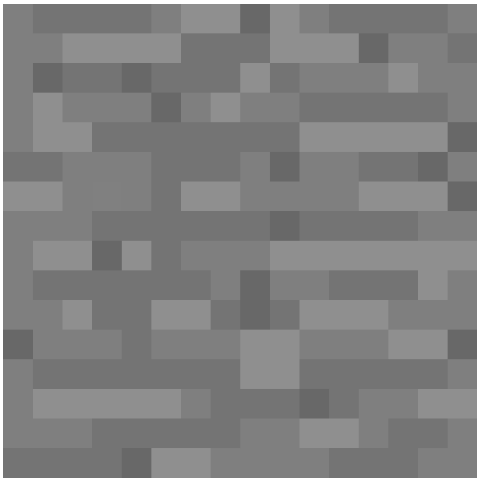

Figure 2: Reading a small 2D texture in nearest

Figure 3: Reading a small 2D texture in linear

In CUDA, texture provide this kind of feature but there are read only or write only. The advantage of surface is there ability to be read and write on the same kernel. Unfortunately, they cannot use the sampling feature describe above. The same memory can be mapped to texture of surface. In function of the use case, texture or surface can be use.