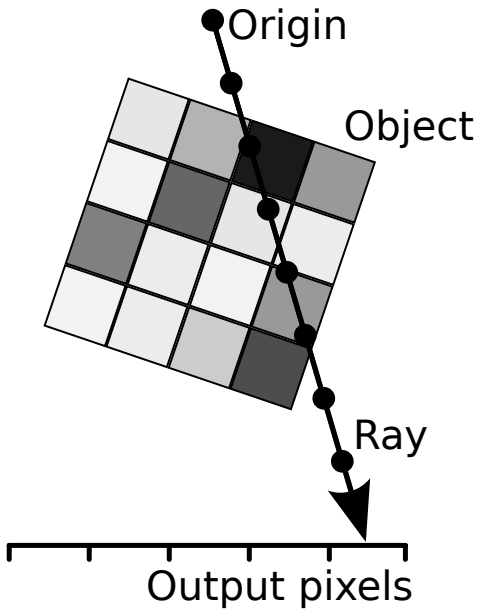
OpenCL exercise 4: Volume rendering

In this exercise a simple volume rendering algorithm should be implemented on the GPU.
For each pixel in the output image, a ray through the object is considered. The sum of all values along this ray is used as color for the pixel in the output image. The step size for the summation is 1. Trilinear interpolation is used for getting values which are not exactly on the grid. Values outside the input object are considered to be zero.



Your task is to:

- Write a GPU implementation of the rendering algorithm. Use a 3D image object for the input data and use samplers to do the interpolation etc.
- Write profiling code which prints the CPU time / GPU time / memory transfer and speedups. For the memory transfer only consider the time for transfering the output data, as the input data only has to be transferred once.
- Try your code with the large data set which on the pool PCs is in /usr/local.nfs/pas/teaching/pas/2014-2/rpi-2.hdf5

Optionally, you can also:

- Use half-precision instead of single precision for the input data on the GPU to reduce memory requirements.
- Use the OpenCL-OpenGL-bridge functionality to avoid having to transfer the output image to the CPU and back to the GPU.