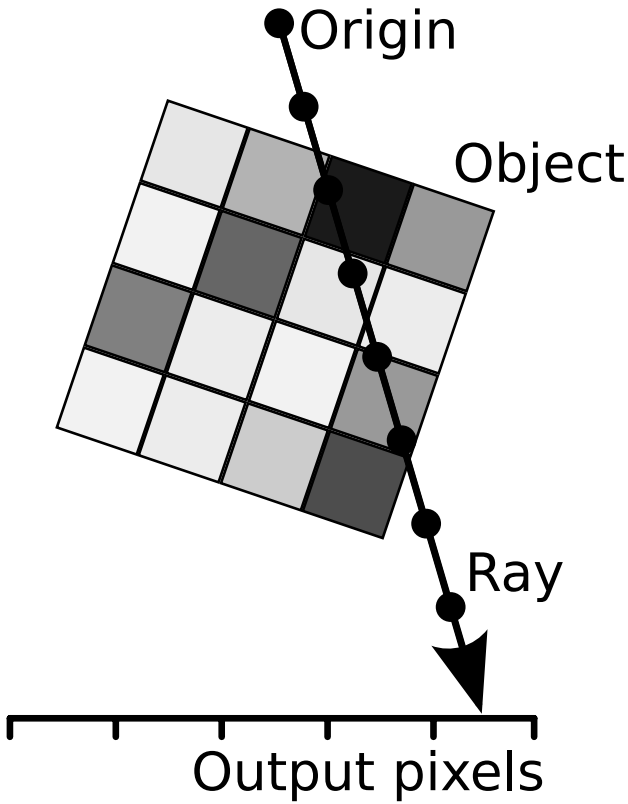
# OpenCL exercise 5: Volume rendering

Kaicong Sun

# Volume rendering




- Ray goes from origin to the output pixels
- Values of object (= input data) along the ray are summed up
- If value is not taken in the middle of a pixel, trilinear interpolation is used (bilinear in 2D-case)
- Sum of the values is value for output pixel
- Values outside the input object = 0

# Task

- GPU implementation of 3D volume rendering
  - Use 3D image object for input data
- Profiling code which prints the CPU time / GPU time / memory transfer and speedups.
  - For memory transfer: Only time for transfering output data
- Try code with large data set

# Hints

```
cl::size_t<3> origin;
origin[0] = origin[1] = origin[2] = 0;
cl::size_t<3> region;
region[0] = countX;
region[1] = countY;
region[2] = countZ;
//For writing 3D image:
queue.enqueueWriteImage(d_input, true, origin, region,
countX * sizeof (float), countX * countY * sizeof (float),
(void*) h_input, NULL, &copyToDev);
//For getting image size:
float3 boxMax = (float3)get_image_width(d_input),
get_image_height(d_input), get_image_depth(d_input)
```