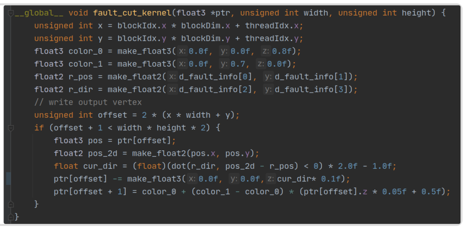
CGT620-Lab5-Zhiquan Wang

Assign	
Status	Not started

CGT620-Lab5-Zhiquan Wang

Setup

https://www.notion.so/zhiquanw/CGT620-Lab5-Zhiquan-Wang-bc35009fa9de4d8b98fa3132a90f9abe



```
__global__ void fault_cut_kernel(float3 *ptr, unsigned int width, unsigned int height) {
    unsigned int x = blockIdx.x * blockDim.x + threadIdx.x;
    unsigned int y = blockIdx.y * blockDim.y + threadIdx.y;
    float3 color_0 = make_float3( x: 0.0f,  y: 0.0f,  z: 0.8f);
    float3 color_1 = make_float3( x: 0.0f,  y: 0.7,  z: 0.0f);
    float2 r_pos = make_float2( x: d_fault_info[0],  y: d_fault_info[1]);
    float2 r_dir = make_float2( x: d_fault_info[2],  y: d_fault_info[3]);
    // write output vertex
    unsigned int offset = 2 * (x * width + y);
    if (offset + 1 < width * height * 2) {
        float3 pos = ptr[offset];
        float2 pos_2d = make_float2(pos.x, pos.y);
        float cur_dir = (float)(dot(r_dir, pos_2d - r_pos) < 0) * 2.0f - 1.0f;
        ptr[offset] -= make_float3( x: 0.0f,  y: 0.0f, z: cur_dir* 0.1f);
        ptr[offset + 1] = color_0 + (color_1 - color_0) * (ptr[offset].z * 0.05f + 0.5f);
    }
}
```

Setup

- Mesh Size: 500 * 500
- Block Size: 16 * 16

Result

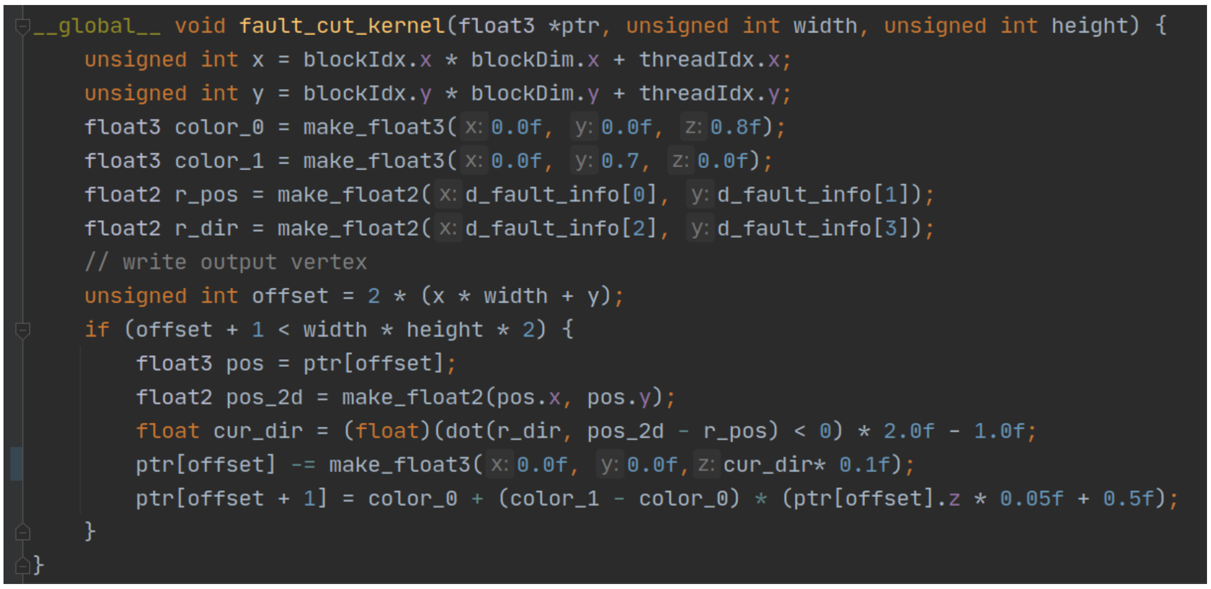
- CPU

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/23090ed0-caca-4cf5-bba9-389807783e44/cpu_000.mp4

- GPU

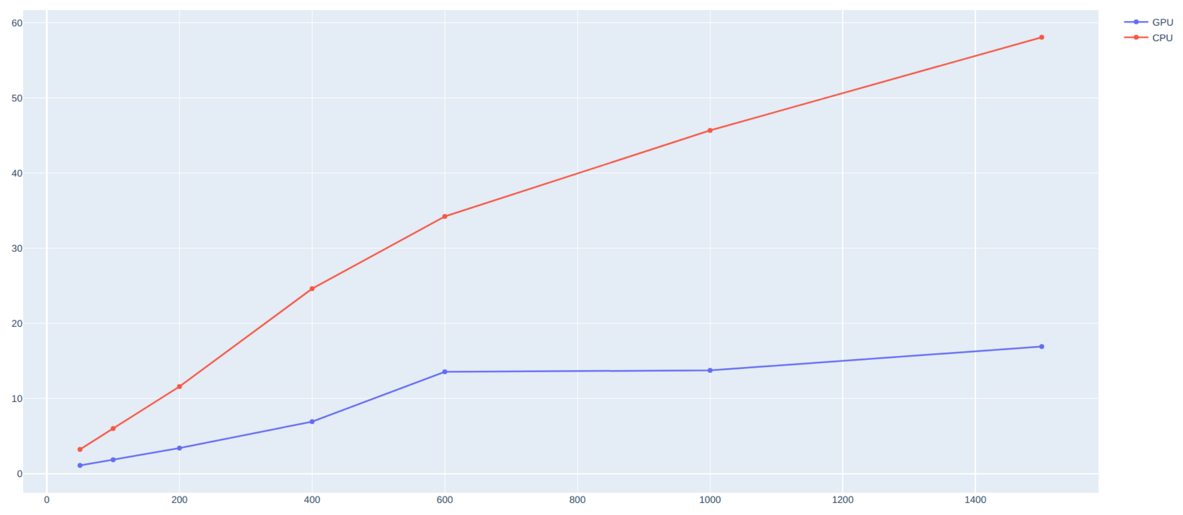
https://s3-us-west-2.amazonaws.com/secure.notion-static.com/8f0d0f54-5827-4d2f-92ab-fd4c4e22207e/gpu_fault.mp4

Implementation (GPU)



```
__global__ void fault_cut_kernel(float3 *ptr, unsigned int width, unsigned int height) {
    unsigned int x = blockIdx.x * blockDim.x + threadIdx.x;
    unsigned int y = blockIdx.y * blockDim.y + threadIdx.y;
    float3 color_0 = make_float3( x: 0.0f,  y: 0.0f,  z: 0.8f);
    float3 color_1 = make_float3( x: 0.0f,  y: 0.7,  z: 0.0f);
    float2 r_pos = make_float2( x: d_fault_info[0],  y: d_fault_info[1]);
    float2 r_dir = make_float2( x: d_fault_info[2],  y: d_fault_info[3]);
    // write output vertex
    unsigned int offset = 2 * (x * width + y);
    if (offset + 1 < width * height * 2) {
        float3 pos = ptr[offset];
        float2 pos_2d = make_float2(pos.x, pos.y);
        float cur_dir = (float)(dot(r_dir, pos_2d - r_pos) < 0) * 2.0f - 1.0f;
        ptr[offset] -= make_float3( x: 0.0f,  y: 0.0f, z: cur_dir* 0.1f);
        ptr[offset + 1] = color_0 + (color_1 - color_0) * (ptr[offset].z * 0.05f + 0.5f);
    }
}
```

Performance



Discussion

The computing time increases linearly as it is a O(n) algorithm.