[MI3.04a] Advanced Programming for HPC

Hello, CUDA!

Labwork 3

HUYNH Vinh Nam

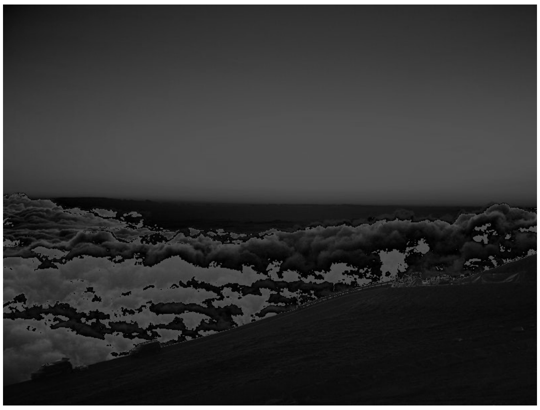
M19.ICT.007

November 2020

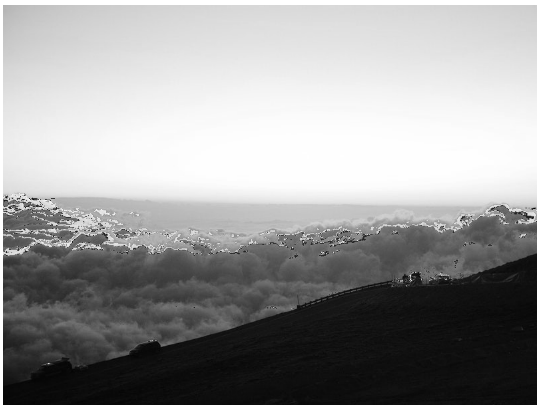
Original Input

(a) Original Image

Outputs



(b) Unwanted Output



(c) Expected Output

Implementation

```
__global__ void grayscale(uchar3 *input, uchar3 *output) {
    int tid = threadIdx.x + blockIdx.x * blockDim.x;
    output[tid].x = (unsigned char)((int)input[tid].x + (int)input[tid].y + (int)input[tid].z) / 3;
    output[tid].z = output[tid].y = output[tid].x;
}

void Labwork::labwork3_GPU() {
    // Calculate number of pixels
    int pixelCount = inputImage->width * inputImage->height;
    char *hostInput = inputImage->buffer;
    outputImage = static_cast<char *>(malloc(pixelCount * 3));

    // Allocate CUDA memory
    uchar3 *devInput;
    uchar3 *devOutput;
    cudaMalloc(&devInput, pixelCount * 3);
    cudaMalloc(&devOutput, pixelCount * 3);

    // Copy CUDA Memory from CPU to GPU
    cudaMemcpy(devInput, hostInput, pixelCount * 3, cudaMemcpyHostToDevice);

    // Processing
    int blockSize = 64;
    int numBlock = pixelCount / blockSize;
    grayscale<<<numBlock, blockSize>>>(devInput, devOutput);

    // Copy CUDA Memory from GPU to CPU
    cudaMemcpy(outputImage, devOutput, pixelCount * 3, cudaMemcpyDeviceToHost);

    // Cleaning
    free(hostInput);
    cudaFree(devInput);
    cudaFree(devOutput);
}
```