

Advaned Programming fo HPC - Report labwork 4

NGUYEN TAT HUNG

November 3, 2021

Implementation

```
void Labwork::labwork4_GPU() {
    // Calculate number of pixels
    int pixelCount = inputImage->width * inputImage->height;
    //char *hostInput = inputImage->buffer; // Perfect version
    char *hostInput = (char*) malloc(inputImage->width * inputImage->height * 3); // Test
    char *hostOutput = new char[inputImage->width * inputImage->height * 3]; // Test version
    outputImage = static_cast<char *>(malloc(pixelCount * 3));

    // Allocate CUDA memory
    uchar3 *devInput;
    uchar3 *devOutput;
    //cudaMalloc(&devInput, pixelCount*3); // Perfect version
    cudaMalloc(&devInput, pixelCount * sizeof(uchar3)); // Test version
    //cudaMalloc(&devOutput, pixelCount*3); // Perfect version
    cudaMalloc(&devOutput, pixelCount * sizeof(float)); // Test version

    // Copy CUDA Memory from CPU to GPU
    //cudaMemcpy(devInput, hostInput, pixelCount*3, cudaMemcpyHostToDevice); // Perfect version
    cudaMemcpy(devInput, hostInput, pixelCount * sizeof(uchar3), cudaMemcpyHostToDevice);

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

    // Copy CUDA Memory from GPU to CPU
    //cudaMemcpy(outputImage, devOutput, pixelCount*3, cudaMemcpyDeviceToHost); // Perfect version
    cudaMemcpy(hostOutput, devOutput, pixelCount*sizeof(float), cudaMemcpyDeviceToHost);

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

Result



Figure 1: Original input image



Figure 2: Output image