[MI3.04a] Advanced Programming for HPC

Threads

Labwork 4

HUYNH Vinh Nam

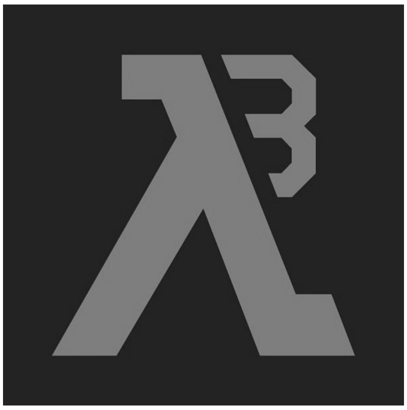
M19.ICT.007

November 2020

Original Input

(a) Original Image

Output



(b) Output

Implementation

```
__global__ void grayscale_2d(uchar3 *input, uchar3 *output) {
    int x = threadIdx.x + blockIdx.x * blockDim.x;
    int y = threadIdx.y + blockIdx.y * blockDim.y;
    int w = blockDim.x * gridDim.x;
    output[x + y*w].x = (char)(((int)input[x + y*w].x + (int)input[x + y*w].y
                              + (int)input[x + y*w].z) / 3);
    output[x + y*w].z = output[x + y*w].y = output[x + y*w].x;
}

void Labwork::labwork4_GPU() {
    int pixelCount = inputImage->width * inputImage->height;
    char *hostInput = inputImage->buffer;
    outputImage = static_cast<char *>(malloc(pixelCount * 3));

    uchar3 *devInput;
    uchar3 *devOutput;
    cudaMalloc(&devInput, pixelCount * 3);
    cudaMalloc(&devOutput, pixelCount * 3);

    cudaMemcpy(devInput, hostInput, pixelCount * 3, cudaMemcpyHostToDevice);

    dim3 blockSize = dim3(32,32);
    dim3 gridSize = dim3(inputImage->width / blockSize.x + 1, inputImage->height / blockSize.y + 1);
    grayscale_2d<<<gridSize, blockSize>>>(devInput, devOutput);

    cudaMemcpy(outputImage, devOutput, pixelCount * 3, cudaMemcpyDeviceToHost);

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