

LABWORK 4

THREAD

TRAN Quy Ban
Department of ICT

Introduction

This labwork tries to improve the speed of the processing by using thread.

Source codes

```
__global__ void grayscaleVer2D(uchar3* input, uchar3* output, int imageWidth, int imageHeight) {
    int tid = threadIdx.x + blockIdx.x * blockDim.x;
    int tidy = threadIdx.y + blockIdx.y * blockDim.y;
    if(tid >= imageWidth || tidy >= imageHeight) return;
    int tid = tid + tidy * imageWidth;
    output[tid].x = (input[tid].x + input[tid].y + input[tid].z) / 3;
    output[tid].z = output[tid].y = output[tid].x;
}
```

```
void Labwork::labwork4_GPU() {
// Calculate number of pixels
    int pixelCount = inputImage->width * inputImage->height;

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

    // // Copy InputImage from CPU (host) to GPU (device)
    cudaMemcpy(devInput, inputImage->buffer, pixelCount * sizeof(uchar3), cudaMemcpyHostToDevice);

    // // Processing : launch the kernel
    // // int blockSize = 1024;
    // // int numBlock = pixelCount / blockSize;
    // // grayscale<<<numBlock, blockSize>>>(devInput, devOutput);
    dim3 blockSize = dim3(32, 32);
    // //dim3 gridSize = dim3(8, 8);
    dim3 gridSize = dim3((inputImage->width + blockSize.x - 1) / blockSize.x, (inputImage->height + blockSize.y - 1) / blockSize.y);
    grayscaleVer2D<>><<<gridSize, blockSize>>>>(devInput, devOutput, inputImage->width, inputImage->height);
}
```

LABWORK

```
    grayscaleVer2D<<<gridSize, blockSize>>>(devInput, devOutput, inputImage->width, i

// // Copy CUDA Memory from GPU to CPU
// // allocate memory for the output on the host
outputImage = static_cast<char *>(malloc(pixelCount * sizeof(uchar3)));
cudaMemcpy(outputImage, devOutput, pixelCount * sizeof(uchar3), cudaMemcpyDeviceToH

// // Cleaning
cudaFree(devInput);
cudaFree(devOutput);

}

}
```

Analysis

1. Improve the labwork

The output image is displayed:

