Report on Labwork 9

TRAN Thi Hong Hanh

November 24, 2019

1 Explain how you implement the labwork?

- Calculate histogram of input **grayscale** image. I used local histogram of a region in the image (GATHER) and then combine all of the region by a sum (REDUCTION) as described on the slide.

```
__global__ void histogramGather(uchar3 *input, unsigned int **output,
                                                    int width, int height)
{
    unsigned int histoL[256] = {0};
    for (int i = 0; i < height; i++)
    {
        int j = input[blockIdx.x * height + i].x;
        histoL[j]++;
    }
    for (int i = 0; i < 256; i++)
    {
        output[blockIdx.x][i] = histoL[i];
    }
}

__global__ void histogramReduction(unsigned int **input, int *output,
                                                    int width, int height)
{
    // Dynamic shared memory size, allocated in host
    __shared__ unsigned int cache[256];

    // Cache the block content
    unsigned int localtid = threadIdx.x;
    cache[localtid] = 0;

    __syncthreads();

    // Reduction in cache
    for (int i = 0; i < width; i++)
```

```
    {
        cache[localtid] += input[i][localtid];
    }
    __syncthreads();

    // Only first thread writes back
    if (localtid == 0)
    {
        for (int i = 0; i < 256; i++)
        {
            output[i] = cache[i];
        }
    }
}
```

- Equalize the histogram for that input image. I took advantage of grayscale in labwork 6, calculate the Cumulative distribution function (CDF), and equalize the histogram by increasing the global contrast and recalculate intensity.

```
__global__ void cdfCalculation(int *h, int pixelCount)
{
    int cdfMin = 0;
    int cdfCumul = 0;
    for (int i = 0; i < 256; i++)
    {
        if (cdfMin == 0)
        {
            cdfMin = h[i];
        }
        cdfCumul += h[i];
        h[i] = round((double)(cdfCumul - cdfMin) / (pixelCount - cdfMin) * 255.0);
    }
}

__global__ void equalization(uchar3 *input, int *h, uchar3 *output,
                                                    int width, int height)
{
    int tidX = threadIdx.x + blockIdx.x * blockDim.x;
    if (tidX >= width)
        return;
    int tidY = threadIdx.y + blockIdx.y * blockDim.y;
    if (tidY >= height)
        return;
    int tid = tidX + tidY * width;
```

```
    unsigned char g = h[input[tid].x];
    output[tid].x = output[tid].y = output[tid].z = g;
}
```

- Command:

```
./labwork 9 ../data/cloud_gray.jpg
```

- Result:

```
USTH ICT Master 2019, Advanced Programming for HPC.
Warming up...
Starting labwork 9
[ALGO ONLY] labwork 9 ellapsed 17.8ms
Labwork 9 ellapsed 26.4ms
```

(a) Original image (b) Equalized

Figure 1: The output image after histogram equalization have a slightly higher contrast compared to the original one.