# GPGPU Assignment #2

TA: Yu Sheng Lin Instructor: Wei Chao Chen

March 10, 2016

## 1 Goals

You have to be creative and make a video in this assignment.

## 2 Requirements

You can try to render Perlin Noise, Simplex Noise, or Stable Fluid. If you want to be more aesthetic, then you can try to render fireworks, ocean or even a short movie.

### 2.1 Video Format

In this lab we will first call `void get_info(Lab2VideoInfo &info)` to get the height $H$, width $W$, fps $N/D$ [1] and number of frames $N_f$ of your video. Then we will call your `void Generate(uint8_t *yuv)` $N_f$ times to get all frames of your video.

Instead of store the pixel RGB values directly, we use the YUV color space, which is much commonly used in video codecs.

$$\begin{cases} Y = +0.299R + 0.587G + 0.114B \\ U = -0.169R - 0.331G + 0.500B + 128 \\ V = +0.500R - 0.419G - 0.081B + 128 \end{cases} \tag{1}$$

Moreover we also subsample U and V channel horizontally and vertically, namely the U and V channel is only $\frac{W}{2} * \frac{H}{2}$[2]. The YUV channels are stored sequentially so the total size of a frame is $1.5WH$.

### 2.2 Sample code

The sample code (Listing 1.) generates a grayscale video which is initially black then becomes bright gradually. "Grayscale" means $R = G = B$, so the YUV is:

$$\begin{cases} Y = +0.299R + 0.587R + 0.114R = R \\ U = -0.169R - 0.331R + 0.500R + 128 = 128 \\ V = +0.500R - 0.419R - 0.081R + 128 = 128 \end{cases} \tag{2}$$

The Y channel is $W * H$, so we first calculate the brightness according to the frame index then fill it to the device pointer.

[1] We use a ratio to represent the fps because we are using the y4m format. If you want 24fps then you shell return $N = 24, D = 1$.

[2] $W$ and $H$ must both be even.

The U and V channel are $\frac{W}{2} * \frac{H}{2}$ each, so we fill 128 to the following memory space. Last we increment the frame index.

```
void Lab2VideoGenerator::Generate(uint8_t *yuv) {
    cudaMemset(yuv, (impl->t)*255/NFRAME, W*H);
    cudaMemset(yuv+W*H, 128, W*H/2);
    ++(impl->t);
}
```

Listing 1: Sample code explanation

The output file is of the y4m raw video format (it may be very large!) and you can use softwares such as Ffmpeg or Avconv to convert it to other compressed formats (Listing 2.).

```
avconv -i output.y4m output.mkv
```

Listing 2: Sample code explanation

Hints: We do not allow you to modify the header, and if you don't know how to achieve that, please read Pimpl Idiom or Opaque Pointer.

# 3 Submission