



Menu Driven Image Manipulation program using cuda

Adithya B [200905382]

Prakhar Choudhary [200905380]

Prodeep Ghosh [200905384]

Introduction

- The "cuda-menu-conversion-to-grayscale-sepia-negative" project provides a command-line tool to convert menu images to grayscale, sepia, or negative using CUDA acceleration. The tool can process JPEG and JPG image formats and save the output as JPG images.
- The project aims to provide a faster and more efficient way to perform image conversions compared to CPU-only implementations by utilizing the power of NVIDIA GPUs and CUDA programming.

Methodology

- The tool uses a CUDA kernel to perform the image conversions. The kernel takes the input image, processes it on the GPU, and outputs the converted image. The kernel implementation includes rescaling the RGB values of each pixel to convert it to grayscale, sepia, or negative. RGB images store 3 values for each cell, the red, green, and blue values, respectively. Grayscale images only store one value for each cell corresponding to how black or white the image is.
- The tool also utilizes multi-threading to process the image data in parallel, improving the processing speed.

- The formula for converting RGB to grayscale is:

```
int gray=0.2126*rgb_image[index]+0.7152*rgb_image[index + 1] + 0.0722 * rgb_image[index + 2];
```
- The formula for converting RGB to sepia is:
 - ```
int sr = (int)(0.393 * r + 0.769 * g + 0.189 * b);
```
  - ```
int sg = (int)(0.349 * r + 0.686 * g + 0.168 * b);
```
 - ```
int sb = (int)(0.272 * r + 0.534 * g + 0.131 * b);
```
- The formula for converting RGB to its negative is:
  - ```
neg_image[index] = 255 - rgb_image[index];
```
 - ```
neg_image[index + 1] = 255 - rgb_image[index + 1];
```
  - ```
neg_image[index + 2] = 255 - rgb_image[index + 2];
```

Negative

```
__global__ void negative(unsigned char  
*rgb_image, unsigned char *neg_image,  
int width, int height) {
```

```
    int row = blockIdx.y * blockDim.y +  
threadIdx.y;
```

```
    int col = blockIdx.x * blockDim.x +  
threadIdx.x;
```



```
if (row < height && col < width) {  
    int index = (row * width + col) * 3;  
    neg_image[index] = 255 - rgb_image[index];  
    neg_image[index + 1] = 255 - rgb_image[index + 1];  
    neg_image[index + 2] = 255 - rgb_image[index + 2];  
}  
}
```

Gray Scale

```
__global__ void grayscale(unsigned char *rgb_image, unsigned char
*gray_image, int width, int height) {
    int row = blockIdx.y * blockDim.y + threadIdx.y;
    int col = blockIdx.x * blockDim.x + threadIdx.x;

    if (row < height && col < width) {
        int index = (row * width + col) * 3;
        int gray = 0.2126 * rgb_image[index] + 0.7152 *
rgb_image[index + 1] + 0.0722 * rgb_image[index + 2];
        gray_image[row * width + col] = (unsigned char) gray;
    }
}
```



sepia

```
__global__ void sepia(unsigned char *rgb_image, unsigned char
*sepia_image, int width, int height) {
    int row = blockIdx.y * blockDim.y + threadIdx.y;
    int col = blockIdx.x * blockDim.x + threadIdx.x;

    if (row < height && col < width) {
        int index = (row * width + col) * 3;
        int r = rgb_image[index];
        int g = rgb_image[index + 1];
        int b = rgb_image[index + 2];
```




```
int sr = (int)(0.393 * r + 0.769 * g + 0.189 * b);
    int sg = (int)(0.349 * r + 0.686 * g + 0.168 * b);
    int sb = (int)(0.272 * r + 0.534 * g + 0.131 * b);
    sepia_image[index] = (unsigned char) (sr > 255 ? 255 : sr);
    sepia_image[index + 1] = (unsigned char) (sg > 255 ? 255 : sg);
    sepia_image[index + 2] = (unsigned char) (sb > 255 ? 255 : sb);
}
}
```

Future Enhancements

There are several potential future enhancements for the tool, such as:

- Adding support for additional image formats
- Providing more conversion options, such as color correction
- Adding a graphical user interface (GUI) for easier use
- Additionally, the tool's performance could be further optimized by using more advanced techniques, such as optimizing memory access patterns and utilizing shared memory.

conclusion

- The "cuda-menu-conversion-to-grayscale-sepia-negative" project provides a useful tool for quickly and efficiently converting menu images to grayscale, sepia, or negative using CUDA acceleration. The tool's use of CUDA programming and multi-threading allows for faster processing times compared to CPU-only implementations.
- With potential future enhancements and optimization, the tool can become even more powerful and useful for various image processing tasks.