

# Grayscale and Blur image

Selene Barrios Cornejo

## 1 REPRESENTACIÓN DE IMAGEN EN COLOR RGB

- Cada píxel de una imagen es un valor RGB
- Los rangos RGB no se distribuyen uniformemente

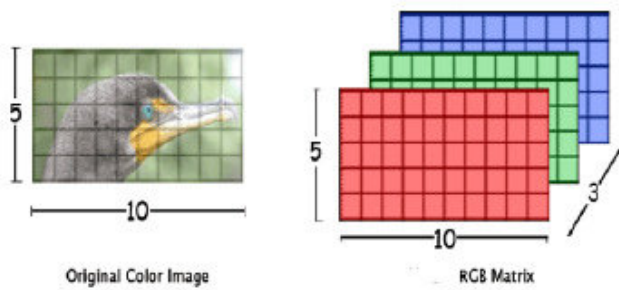


Figure 1: El formato de la fila de una imagen es (r g b) (r g b) ... (r g b)

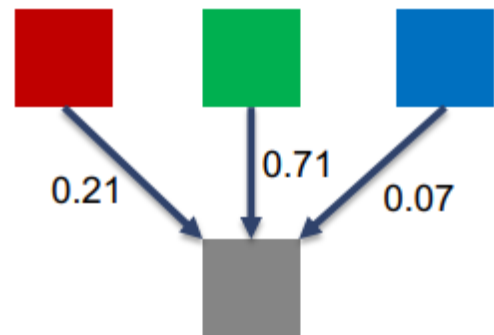


Figure 2: Esto es solo un producto escalar  $\langle [r,g,b], [0.21, 0.71, 0.07] \rangle$  con constantes específicas para el espacio RGB de entrada

## 2 CONVERSIÓN DE RGB A ESCALA DE GRISES

Una imagen digital en escala de grises es una imagen en la que el valor de cada píxel contiene solo información de intensidad.

### 2.1 Fórmula de cálculo de color

- Para cada píxel (r g b) en (I, J) haz lo siguiente:  $\text{píxelgris}[I,J] = 0,21*r + 0,71*g + 0,07*b$

- Barrios Cornejo Selene  
sbarrios@unsa.edu.pe,  
Universidad Nacional de San Agustín.

## 3 CÓDIGO DE CONVERSIÓN DE RGB A ESCALA DE GRISES

```
#define CHANNELS 3 // tenemos 3 canales
                    // correspondientes a RGB
// La imagen de entrada esta codificada
// como caracteres [0, 255]
__global__ void colorConvert(unsigned
char * grayImage,
unsigned char * rgbImage,
int width, int height) {
    int x = threadIdx.x + blockIdx.x *
        blockDim.x;
    int y = threadIdx.y + blockIdx.y *
        blockDim.y;
    if (x < width && y < height) {
        // obtener coordenadas 1D para la
        // imagen en escala de grises
        int grayOffset = y*width + x;
        // se puede pensar que la imagen RGB
        // tiene
```

```
// CANAL multiplicado por columnas
// que la imagen en escala de grises
int rgbOffset = grayOffset*CHANNELS;
unsigned char r = rgbImage[rgbOffset
]; //valor rojo por pixel
unsigned char g = rgbImage[rgbOffset
+ 2]; //valor verde por pixel
unsigned char b = rgbImage[rgbOffset
+ 3]; //valor azul por pixel
// realizar el cambio de escala y
guardarlo
// Multiplicamos por constantes de
punto flotante
grayImage[grayOffset] = 0.21f*r +
0.71f*g + 0.07f*b;
}
}
```

## 4 IMAGEN BORROSA

Reducir o distorsionar los detalles de una imagen.

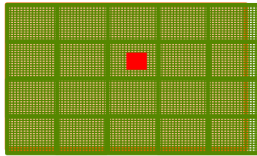


Figure 3: Píxeles procesados por un bloque de hilos

### 4.1 Desenfoque de imagen como kernel 2D

```
__global__
void blurKernel(unsigned char * in,
unsigned char * out, int w, int h) {
int Col = blockIdx.x * blockDim.x +
threadIdx.x;
int Row = blockIdx.y * blockDim.y +
threadIdx.y;
if (Col < w && Row < h) {
int pixVal = 0;
int pixels = 0;
// Obtener el promedio del cuadro
circundante de 2xBLUR_SIZE x 2
xBLUR_SIZE
for(int blurRow = -BLUR_SIZE; blurRow
< BLUR_SIZE+1; ++blurRow) {
for(int blurCol = -BLUR_SIZE;
blurCol < BLUR_SIZE+1; ++
blurCol) {
int curRow = Row + blurRow;
int curCol = Col + blurCol;
```

```
// Verificamos que tengamos
un pixel de imagen
v lido
if(curRow > -1 && curRow < h
&& curCol > -1 && curCol <
w) {
pixVal += in[curRow * w +
curCol];
pixels++; // Mantenemos
un registro del numero
de pixeles en el
total acumulado
}
}
// Escribe nuestro nuevo valor de
pixel
out[Row * w + Col] = (unsigned char) (
pixVal / pixels);
}
}
```

## 5 RESULTADOS

El siguiente resultado lo obtuve en el siguiente [COLAB](#) ya que en mi local no logre ejecutar el código de manera correcta.

```
(base) pejelagarta@pejelagarta-FX503VD: ~/Desktop/PARELELA/CUDA-Training/cudaExercises/chapter03$ nvcc -o executable colorToGrayscale.cu
/usr/bin/ld: /tmp/tmpxft_00004204_00000000-11_colorToGrayscale.o: in function 'lodepng_decode32_file(char const*)':
tmpxft_00004204_00000000-6_colorToGrayscale.cudafe1.cpp:(.text+0x25d): undefined reference to 'lodepng_decode32_file(unsigned char**, unsigned int*, unsigned int*, char const*)'
/usr/bin/ld: tmpxft_00004204_00000000-6_colorToGrayscale.cudafe1.cpp:(.text+0x270): undefined reference to 'lodepng_error_text(unsigned int)'
/usr/bin/ld: /tmp/tmpxft_00004204_00000000-11_colorToGrayscale.o: in function 'lodepng_encode32(char const*, unsigned char*, int, int)':
tmpxft_00004204_00000000-6_colorToGrayscale.cudafe1.cpp:(.text+0x2d6): undefined reference to 'lodepng_encode32_file(char const*, unsigned char const*, unsigned int, unsigned int)'
/usr/bin/ld: tmpxft_00004204_00000000-6_colorToGrayscale.cudafe1.cpp:(.text+0x2e9): undefined reference to 'lodepng_error_text(unsigned int)'
collect2: error: ld returned 1 exit status
(base) pejelagarta@pejelagarta-FX503VD: ~/Desktop/PARELELA/CUDA-Training/cudaExercises/chapter03$
```

Figure 4: Error que obtengo al intentar ejecutar colorToGrayscale.cu en CUDA



Figure 5: Conversión de RGB a GrayScale

Use una imagen en escala de grises como input para blur.cu

```
pejelagarta@pejelagarta-FX503VD: ~/Desktop/PARALELA/C...  
stb_image.h(5121): warning #550-D: variable "idata_limit_old" was set but never  
used  
stb_image.h(6894): warning #550-D: variable "out_size" was set but never used  
stb_image.h(6895): warning #550-D: variable "delays_size" was set but never used  
(base) pejelagarta@pejelagarta-FX503VD:~/Desktop/PARALELA/CUDA-Training/cudaExer  
cises/chapter0: $ ./blur  
(base) pejelagarta@pejelagarta-FX503VD:~/Desktop/PARALELA/CUDA-Training/cudaExer  
cises/chapter0: $ nvcc -o executable blur.cu  
stb_image.h(4207): warning #550-D: variable "old_limit" was set but never used  
stb_image.h(5121): warning #550-D: variable "idata_limit_old" was set but never  
used  
stb_image.h(6894): warning #550-D: variable "out_size" was set but never used  
stb_image.h(6895): warning #550-D: variable "delays_size" was set but never used  
(base) pejelagarta@pejelagarta-FX503VD:~/Desktop/PARALELA/CUDA-Training/cudaExer  
cises/chapter0: $ ./blur  
(base) pejelagarta@pejelagarta-FX503VD:~/Desktop/PARALELA/CUDA-Training/cudaExer  
cises/chapter0: $
```

Figure 6: Ejecución del código de blur en CUDA



Figure 7: Conversión de GrayScale a Blur

## 6 REPOSITORIO

El código se encuentra disponible en el siguiente repositorio de [GITHUB](#)