

Relatório de Arquitetura de Computadores e Sistemas

Pedro Mendes nº37770

Ruben Peixoto nº37514

25 de Maio de 2017

1 Introdução

Este trabalho consiste na criação de um programa em que, ao ser introduzido uma imagem em RGB, devolve uma imagem com contornos obtidos a partir da aplicação dos operadores Sobel Horizontal e Vertical. Com isto dividimos o trabalho em 3 funções, uma que irá converter a imagem RGB e devolve a imagem com tons de cinzentos, outra função que se aplica as matrizes Sobel e outra função que junta as imagens em que se aplicou as matrizes Sobel. Porém o grupo não concluiu o trabalho com sucesso realizando somente as funções gray e contour. Embora não tenhamos testado a ultima função.

1.1 Função Gray

A função gray irá passar a imagem de rgb para gray, para tal, o grupo teve que pegar um pixel de cada vez (cada pixel é composto por 3 bytes pelo Red, Green e Blue) multiplicou o byte Red por 30 depois, multiplicou-se o byte Green por 59 e para o byte Blue multiplicamos por 11. Após estas operações somamos os três bytes obtidos e dividimos por 100. Com o resultado obtido, colocamos o respetivo valor no primeiro byte do bufferGray.

Porém neste processo tivemos alguns problemas:

- Escolher a melhor forma agrupar os três bytes (Red, Green e Blue) e realizar as repetivas operações;
- Tivemos problemas sobre como guardar alocar espaço tanto para o bufferRGB como o bufferGRAY;

1.2 Função Convolution

Embora o grupo não tenha conseguido realizar a função convolution ,pensamos, para esta função, em começar a aplicar as matrizes sobel a partir do segundo elemento da segunda linha da imagem gray original e ao guardar num novo buffer iria preencher as margens da imagem obtida com zeros. Contudo surgiu-nos os seguintes problemas:

- Colocar as margens; —item Aplicar as matrizes Sobel Horizontal e Vertical;

1.3 Função Contour

Para a função contour usamos os buffer que continham as imagem multiplicadas com as matrizes Sobel (Horizontal e Vertical) criou-se um ciclo para realizar esta função. Para isso pegamos em cada byte (um de cada vez) dividimo-los por 4 , realizamos a operação uma operação OR com os bytes em questão e guardamos num registo. Após obtermos esse resultado divimos por 2 e subtraímos esse valor a 255. De início houve algumas dificuldades:

- De início pensamos em colocar três ciclos, um para dividir os buffer Sobel por 4, outro para realizar uma operação OR entre os buffer e outro ciclo para subtrair 255 a cada byte da imagem;