



UNIVERSIDADE DE ÉVORA

Relatório Trabalho - Redução de Ruído

Miguel de Carvalho - 43108

Ricardo Oliveira - 42647

5 de junho de 2019

Resumo

Este trabalho tem como objetivo a redução de ruído apresentado numa imagem em tons de cinzento, através de dois métodos distintos, a média e a mediana.

1 Introdução

Com a realização deste trabalho pretende-se desenvolver um programa em assembly MIPS capaz de reduzir algum do ruído apresentado numa imagem em tons de cinzento. Para tal, foram implementados dois métodos distintos, a média e a mediana.

Após a leitura do enunciado, conseguimos compreender como as imagens são constituídas pelas cores RGB, do mesmo modo, percebemos que as imagens em escala de cinzentos têm o mesmo valor nas três componentes RGB.

Na implementação dos dois métodos, encaramos algumas dificuldades na interpretação do enunciado e a iniciar a implementação das funções em assembly. Inicialmente, o programa deve abrir o ficheiro que contem a imagem, já convertida, em formato '.gray'. Após a leitura da imagem, aplica-se um dos dois filtros que, após a execução do seu algoritmo, irá proceder à escrita do resultado obtido. Finalmente, o programa irá fechar ambas as imagens (original e filtrada).

2 Implementação

Iniciamos o trabalho com a alocação de espaço para os Buffers (BUFFER A, BUFFER B e BUFFER C), um para a **imagem original**, outro para a **imagem filtrada** e finalmente para a **matriz auxiliar** (para a ordenação dos bits), respetivamente. Procedemos à criação de um menu com três opções de escolha. O número '1' para a média, o número '2' para a mediana, o número '3' para sair do programa, realizado através de syscalls.

2.1 Média

O filtro de média percorre o Buffer que contem a imagem original (BUFFER A) substituindo cada pixel, pela média dos pixels em seu redor. Porém, este não é o melhor filtro a ser aplicado pois, tal como é referido no enunciado, para além de remover algum ruído, esbatesse a própria imagem. Isto deve-se ao facto de, à medida que se percorre a imagem os pixels vão sendo substituídos pela média dos pixels em seu redor (como já foi referido anteriormente), portanto os pixels que não continham ruído, vão sempre ser afetados pelos que continham. A média é calculada e substituí o pixel na mesma posição, mas por sua vez no BUFFER B.

2.2 Mediana

O filtro da mediana percorre o Buffer que contem a imagem original (BUFFER A) substituindo cada pixel, pela mediana dos pixels em seu redor. Por sua vez, este filtro, ao contrário do filtro de média, é mais eficiente e não esbate a imagem, dando assim uma qualidade superior à mesma. O algoritmo organiza os pixels por ordem crescente num Buffer auxiliar (BUFFER C), visto que são nove pixels a mediana será sempre o valor central, esse valor irá substituir o pixel correspondente à posição à qual se está a percorrer o BUFFER A, substituindo-o na mesma posição do BUFFER B.

3 Resultados

Ao realizar a função do filtro de média, recorremos à criação de quatro ciclos 'for', o que não resultou devidamente, ficando o programa a correr num ciclo infinito. Após este equívoco, verificámos assim que seria melhor percorrer os Buffers apenas com um ciclo 'for'. De modo preliminar a *Stack* foi alojado de forma incorreta, visto que deslocamos o *StackPointer* para a zona do Sistema Operativo e assim acionado um comando que emitiu uma mensagem de erro '*address out of range*'. Após a correção dos erros acima referidos, obtivemos o resultado esperado.

No entanto, para o filtro de mediana, presumimos que o raciocínio pretendido é o que temos, porém não obtivemos o *output* esperado, concluindo que o erro tanto pode ser de código ou lógico.



*Imagem à esquerda - output do Filtro de Média. Imagem à direita - output do Filtro de Mediana

4 Conclusão

Em suma, este trabalho fez-nos adquirir ainda mais conhecimentos na linguagem de assembly MIPS, aprendendo a fazer um programa numa linguagem onde devemos encarar o objetivo pedido e todo o processo de forma diferente, visto que, por exemplo, os ciclos são elaborados de forma diferente.