Relatório de Arquitetura de Sistemas e Computadores I

Dinis Matos - 42738
Jose Santos - 43017

9 de Junho de 2019

1 Introdução

Este trabalho tem como objectivo remover o ruido de uma imagem em tons de cinzento, utilizando um conjunto de funções em assembly. Deverá abrir a imagem, correr o algoritmo de remoção de ruido e guardar o resulado noutro ficheiro com o mesmo formato e dimensões.

2 Funções

Foram utilizadas as seguintes funções

2.1 main: São chamadas as funçoes, e inicializa os argumentos das mesmas.

2.2 Read_gray_image: Abre e lê a imagem, Memoria1 (buffer) irá conter a imagem original e a Mermoria2 (buffer) irá conter a imagem filtrada, por fim fecha a imagem.

2.3 Write_gray_image: Começa por alocar espaço na pilha, abre a imagem para escrever, escreve e fecha.

2.4 mean_filter: Começa por alocar espaço na pilha, para cada pixel fora das laterais, é calculada a média dos valores de grey à volta, incluindo o pixel central, e altera o valor do pixel central, isto até todos os pixeis serem alterados.

2.5 median_filter: Para cada pixel fora das laterais, é calculada a mediana dos valores de grey à volta, incluindo o pixel central, e altera o valor do pixel central, isto até todos os pixeis serem alterados (guardado tudo na Memoria2).

2.6 ciclo, ciclo2: Esta função servirá para utilizar os pixeis seguintes (recursividade).

2.7 ciclox: Carrega os pixeis nas variáveis temporárias, soma-se os valores de cada pixel e divide-se por 9.Depois irá verificar se a linha acabou, senão, repete o ciclo, se sim, irá verificar se todos os bytes foram lidos. Se todos os bytes não forem lidos repete o ciclo, senão acabava a função.

2.8 ciclox2 Creega os pixeis nas variáveis temporárias, e adiciona ao endereço s6 o numero de ciclos.

2.9 salto1-8: Comparam os valores de grey de 2 pixeis, ordenando-os de forma a calcular a mediana. No salto 8, após realizar a comparação verifica se o ciclo das comparações acabou, se sim, guarda a mediana no Memoria2, senão, corre o ciclo novamente. Depois irá verificar se a linha acabou, senão, repete o ciclo, se sim, irá verificar se todos os bytes foram lidos. Se todos os bytes não forem lidos repete o ciclo, senão acabava a função.

3 Conclusão

Concluimos que apesar de nao termos obtido o melhor resultado possivel, este projeto ajudou-nos a entender melhor a linguagem de programação Assembly da arquitetura MIPS.
Nota: no calculo da mediana em vez de usarmos o t4 (o valor medio), utilizamos o t6, pois este produzia um melhor resultado.