

Relatório do Trabalho Prático

Arquitetura de Sistemas e Computadores I

Leonel Corado (42602)
Tomás Dias (42784)

Docente: Miguel Barão

9 de Junho de 2019



1 Introdução

Este trabalho tem como objetivo remover ruído de uma imagem de teste Lena aplicando 2 filtros: filtro de média e filtro de mediana. O filtro de média consiste em substituir cada pixel pela média dos pixels numa vizinhança à sua volta. Como pixels vizinhos têm tipicamente intensidades parecidas, a média tem o efeito de substituir os pixels de ruído por valores semelhantes aos vizinhos.

O filtro de mediana é semelhante ao filtro de média, mas em vez de substituir cada pixel pela média da vizinhança, substitui pela mediana. Os valores da matriz são ordenados de menor para maior e é depois selecionado o valor do meio.

Em primeiro lugar, estudou-se atentamente a parte teórica do enunciado cedido pelo professor em que de forma resumida continha algumas noções sobre: cores RGB, tons de cinzento, conversão de RGB para tons de cinzento, o funcionamento da representação de imagens monocromáticas e de imagens RGB, formatos de imagem e conversão de ficheiros e o funcionamento do processamento de imagem. Com estas noções bem assimiladas, estudou-se mais aprofundadamente o modo como o processamento da imagem seria feito, nomeadamente os filtros que seriam necessários aplicar para remover e corrigir o ruído das imagens. De seguida, tentou-se compreender de que forma a implementação desses filtros seria estruturada e quais as funções seriam usadas para o efeito. Essas funções serão definidas mais à frente. Com o enunciado completamente estudado, e com a noção de como o código seria efetuado, passou-se ao desenvolvimento do mesmo.

2 Implementação

mean_filter:

Depois de entender como uma imagem funcionava, concluímos que a informação da imagem estava num buffer, um array 1D. Para procurar os valores da vizinhança de certo byte, concluímos que bastava só buscar -513, -512, -511, etc. posições em relação ao byte que vamos escrever. Em relação aos bounds do array, para que não fosse fora do array, decidimos começar do byte 513 até 512x512 - 513.

median_filter:

Para o filtro de mediana, o mesmo algoritmo do filtro de média para procura de bytes foi utilizado, só que depois, para fazer o store do byte para output, antes era aplicado um algoritmo de sort. Os valores que foram carregados era metidos num array, e, quando chegava ao algoritmo, esse mesmo percorria o array, e fazia um algoritmo de sort para ordenar esses valores no array. No fim, depois de feito o algoritmo, o byte do meio do array, ou seja, o byte 5, era escolhido e posto no output buffer.

3 Conclusão

Em relação ao trabalho, só foi possível fazer o output aplicando o filtro de média. No filtro de mediana, era sempre apresentado uma exceção *Address out of range* quando fazíamos load byte unsigned a um valor guardado no array. Foi experimentado realizar o algoritmo utilizando a stack, mas como a stack só funciona com 32-bit words (logo ocupam 4 bytes), não era muito efetivo utilizar esta mesma. Podemos concluir que apesar do trabalho não apresentar os resultados cem por cento corretos, pensamos que o que fizemos foi satisfatório em relação ao que foi pedido.