

Trabalho prático I

João Gabriel Padovani Bellon

¹Universidade Tecnológica Federal do Paraná (UTFPR)
Departamento Acadêmico de Informática (DAINF)

²Estruturas de Dados 2

1. Definição do Trabalho

Esse trabalho prático tem por objetivo o estudo e implementação de um algoritmo por força bruta e um por divisão em conquista para a resolução de um mesmo problema. Foi escolhido o algoritmo de remoção de ruído *salt and pepper*, que consiste em remover pixels pretos e brancos na imagem, que formam ruído. Foram implementadas duas versões do algoritmo, uma aplicando o método via força bruta e a outra utilizando recursão como método de divisão e conquista.

A imagem padrão utilizada foi a seguinte:



Imagem com ruído *salt and pepper*

A imagem utilizada tem 512x512 pixels e está no formato PGM P2, ou seja, em escala de cinza, com essa escala variando de 0 a 255, branco e preto respectivamente.

2. Algoritmo de Remoção de *Salt and Pepper Noise*

O algoritmo consiste em carregar uma imagem com ruído, então popular uma matriz bidimensional com os dados da imagem. Após, a matriz deve ser passada para uma função responsável por percorrer-la e, em cada item aplicar uma função que irá retirar mediana dela e de seus vizinhos.

A mediana é retirada colocando o elemento da posição i, j e seus vizinhos $i + k, j + k$, sendo $-1 \leq k \leq 1$ em um novo vetor e o ordenando.

$i - 1, j - 1$	$i, j - 1$	$i + 1, j - 1$
$i - 1, j$	i, j	$i + 1, j$
$i - 1, j + 1$	$i, j + 1$	$i + 1, j + 1$

Vizinhos de i, j

Após ordenado, a mediana é selecionada e substituída na posição i, j da matriz. Podem ocorrer casos na qual a mediana se torna o próprio pixel preto (valor 255) ou branco (valor 0) e para estes casos, aplicamos mais uma comparação para fazer a checagem.

3. Resultados

Após a execução do algoritmo na imagem, podemos ver que a limpeza da imagem não é perfeita e que há uma significação de tempo na execução dos métodos.



Imagem após força bruta



Imagem após recursivo

Além dos resultados da imagem, podemos ver as diferenças de se usar os métodos de força bruta e divisão em conquista, aliados aos algoritmos de ordenação corretos.

Método	Tempo (Segundos)
FORÇA BRUTA (Bubble Sort)	0.027600
FORÇA BRUTA (Quick Sort)	0.013060
RECURSÃO (Bubble Sort)	0.027579
RECURSÃO (Quick Sort)	0.011927

Apesar de a imagem utilizada neste trabalho ser pequena, podemos notar que a diferença de tempo na execução dos métodos, pode ser um exemplo de como a escolha e boa implementação de algoritmos impacte no desempenho em aplicações maiores.