

UNIVERSIDADE FEDERAL DE SANTA CATARINA – UFSC- CTC  
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO  
PROJETO E ANÁLISE DE ALGORITMO – 2017-2

Prof. Alexandre Gonçalves Silva

Aluno: Osmar de Oliveira Braz Junior

### Filtro de Mediana

Implemente duas versões do **filtro de mediana**, considerando os dois algoritmos desenvolvidos no item (a), para matrizes bidimensionais  $m \times n$  de inteiros  $0 \leq f(i, j) \leq 255$ , sendo  $0 \leq i < m$ ,  $0 \leq j < n$ , supondo janela de filtro com vizinhança parametrizável de  $p \times q$ , sendo  $2 \leq p < m$  e  $2 \leq q < n$ . A técnica, exemplos e código (em C) podem ser consultados no seguinte documento: <https://www.ime.usp.br/~reverbel/ccm118-12/eps/ep4.pdf>.

Avalie o tempo de execução real (por exemplo, em segundos) das duas versões implementadas do filtro para uma matriz (imagem) suficientemente grande ( $\geq 640 \times 480$  pixels) e para diferentes escolhas de  $p$  e  $q$  (por exemplo, 3, 7, 15, ...).

R.:

O programa foi desenvolvido em Java para desktop e permite escolher quadros de  $p$  por  $q$  de tamanhos 3, 5 e 16. Além de permitir escolher qual o filtro de mediana a ser aplicado na imagem. Depois de escolher os parâmetros basta clicar no botão processar para executar o filtro nas imagens do diretório padrão. O usuário pode selecionar uma imagem a ser processada clicando no botão Abrir Imagem e depois clicando no botão processar. Após o processamento duas janelas serão exibidas com a imagem original e a imagem processada pelo filtro. O programa pode ser executado pelo arquivo **Principal.java**.



As estatísticas são apresentadas no console de execução.

```

C:\windows\system32\cmd.exe
*** Executando projeto ***
-----
Estatísticas da execução do Filtro da Mediana
Nome da Imagem: ImagemCormenRuido.jpg Tipo da Imagem: 5
Tamanho da Imagem: Colunas 1000 Linhas 750
Quadro pxq: 3 x 3
Processada com filtro de mediana(Uersão 1<nlogn>) em 669 milisegundos
-----
Estatísticas da execução do Filtro da Mediana
Nome da Imagem: ImagemCormenRuido.jpg Tipo da Imagem: 5
Tamanho da Imagem: Colunas 1000 Linhas 750
Quadro pxq: 7 x 7
Processada com filtro de mediana(Uersão 1<nlogn>) em 4331 milisegundos
-----
Estatísticas da execução do Filtro da Mediana
Nome da Imagem: ImagemCormenRuido.jpg Tipo da Imagem: 5
Tamanho da Imagem: Colunas 1000 Linhas 750
Quadro pxq: 15 x 15
Processada com filtro de mediana(Uersão 1<nlogn>) em 40150 milisegundos

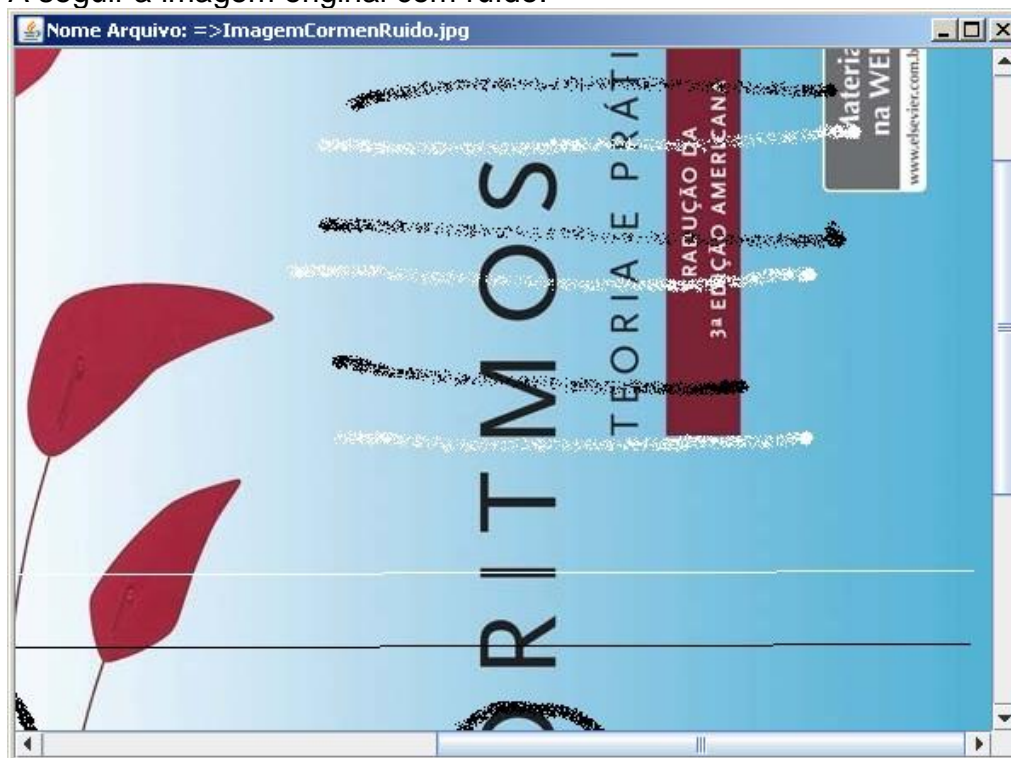
```

Foram realizados testes e estes são apresentados no quadro a seguir

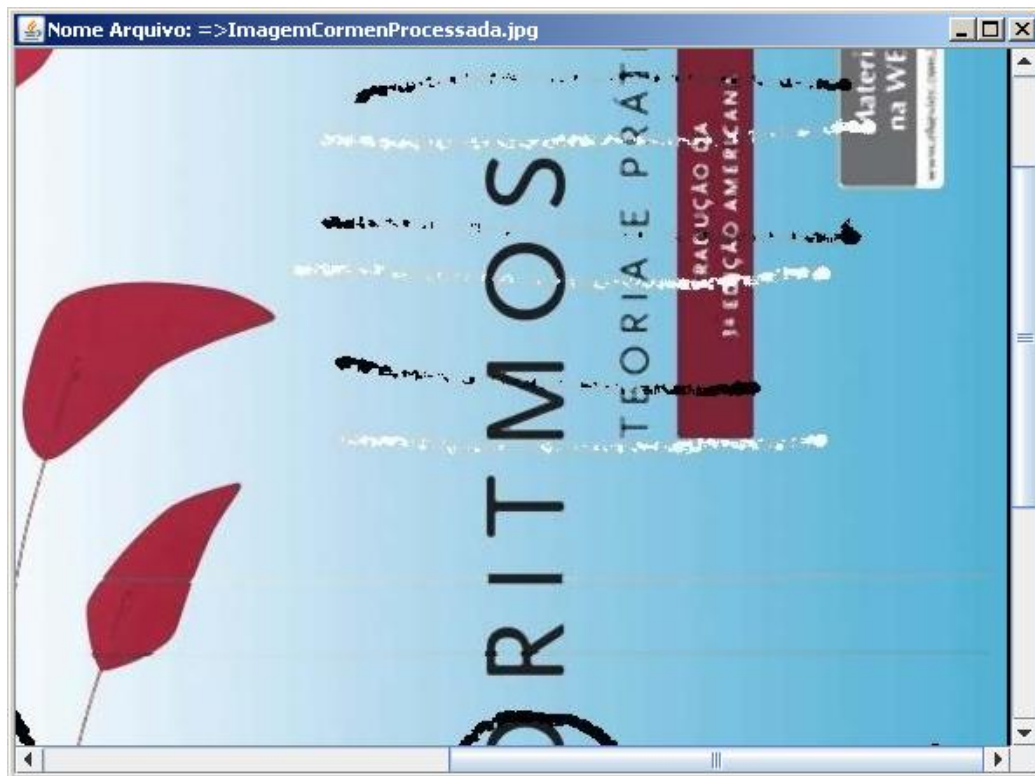
Imagem ColunaxLinha	Filtro	Quadro pxq	Tempo
1000x750	n log n	3x3	635 milissegundos
1000x750	n log n	7x7	4353 milissegundos
1000x750	n log n	15x15	28180 milissegundos
1000x750	n	3x3	1134 milissegundos
1000x750	n	7x7	4867 milissegundos
1000x750	n	15x15	37530 milissegundos

Resultado na imagem do filtro n log n sobre a imagem:

A seguir a imagem original com ruído:



E o resultado da imagem processada



O filtro da mediana está sendo aplicado nos 3 canais de cores da imagem (rgb). O código fonte que percorre a imagem aplicando o filtro da mediana está apresentado a seguir:

```
// Processa valores da imagem de entrada e armazena na imagem de
saida
for(int i=(p/2); i < (alturaImagem - (p/2)); i++){
    for(int j=(q/2); j < (larguraImagem - (q/2)); j++){
        for(int canal=0;canal<=2; canal++){
            //Conta os elementos inseridos no vetor
            int x = 0;
            //Vetor base para achar a mediana
            int V[] = new int[p*q];
            //Recupera os elementos para o quadro
            for (int k = i - (p/2); k <= i + (p/2); k++) {
                //Calcula o inicio do intervalo
                int inicio = j - (q/2);
                //Calcula o fim do intervalo
                int fim = j + (q/2) + 1;
                for (int elemento = inicio; elemento < fim;
elemento++){
                    V[x]=raster.getSample(elemento,k,canal);
                    x = x + 1;
                }
            }
            int t = 0;
            //Seleciona o tipo do filtro da mediana
            if (tipoFiltro==0){
                t = medianaNLogN(V,0,V.length);
```

```
        } else {
            t = medianaN(V,0,V.length);
        }
        int mediana = V[t];
        //Salva a mediana na imagem de saida
        wraster.setSample(j,i,canal,mediana);
    }
}
}
```