**Project number:**

**Course name:** FTL079 – PROCESSAMENTO DIGITAL DE IMAGENS

**Student's name:**

**#1: Diego Giovanni de Alcântara Vieira**

**#2: Lucas Lima de Oliveira**

**Date due: 01 de novembro de 2020**

**Date handed in: 01 de novembro de 2020**

**Technical discussion and results**

1. **Introdução**

Este projeto de trabalho têm como finalidade a implementação de duas funções com diferentes técnicas de filtragem espacial. Sendo assim, foi desenvolvido uma função f = mainProj6.m no *software* Matlab para validação das funções criadas e apagando os testes solicitados no script do projeto. Criado uma função g = twodSFilter.m para filtragem espacila de kernel 2D para máscaras 3x3, 11x11 e 21x21. E a função f = medianSFilter.m para filtragem espacial 2D em mediana, Nessa função ocorre o padding da imagem pela técnica replicate.

1. **Discussão Técnica do Trabalho**

No início da implementação do script no software MatLab usamos a função g = medianFilter(f,3) para o padding da imagem onde f é a imagem utilizada para testes, r = 3 é tamanho do kernel que é igual ao n° de colunas de padding. Ademais, utilizamos a funcão g = medianSFilter(f,w) para transformer cada vizinhança de pixel em um único vetor coluna e usar todos esses vetores de coluna para criar uma única matriz. Cada coluna representaria a intensidade de pixel dentro da vizinhança de pixel. Em seguinda é feita uma classificação ao longo das linhas de cada coluna, pegamos em seguida o vetor do meio da matriz gerada que representa o valor da mediana do filtro de pixel. Por fim, remodelamos a imagem desse vetor ao mesmo tamanho original.

Na primeira aplicação do *script* criamos a função g = medianFilter(f,3) para o *kernel* da imagem de teste. Usamos diversas operações para captura da vizinhanças de pixel, para isso aplicamos o *padding* na imagem em seguida usamos o filtro de média.

Nessa segunda parte, esperamos que a filtragem de mediana mantivesse as bordas bem mantidas ao fazer a suavização da imagem. Como o *script* da figura 2 obtemos o desejado na simulação.

Para o Segundo filtro implementa a filtragem espacial por 2D por Kernel em diferentes tamanhos de mascaras. Utilizamos no início do *script* um função g = twodFilter(f,w) , onde f é a imagem a ser utilizada e w é o neighborhood size.Aplicando os padding na imagem em seguida a aplicação do filtro para diferentes mascaras. Como segue:

1. **Resultados**

**3.1 Função twodSFilter.m**

A função twodFilter.m implementa um filtro de espacial por 2D por diferentes kernel. O *padding* é feito automaticamente no corpo da função conforme solicitado no document de requisites do projeto. Como segue:

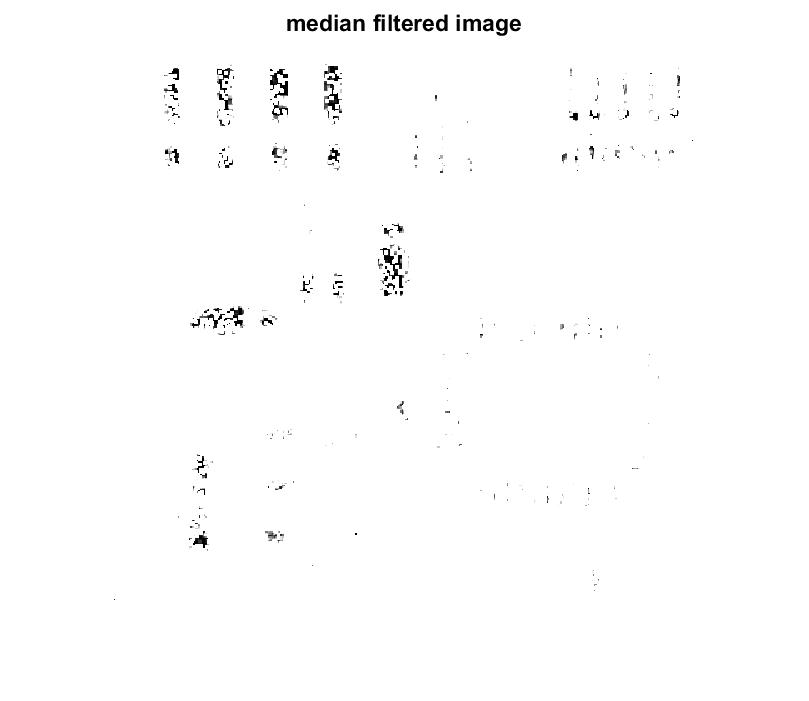


Figura 4: Implementação da filtragem median 2D por kernel 3x3.

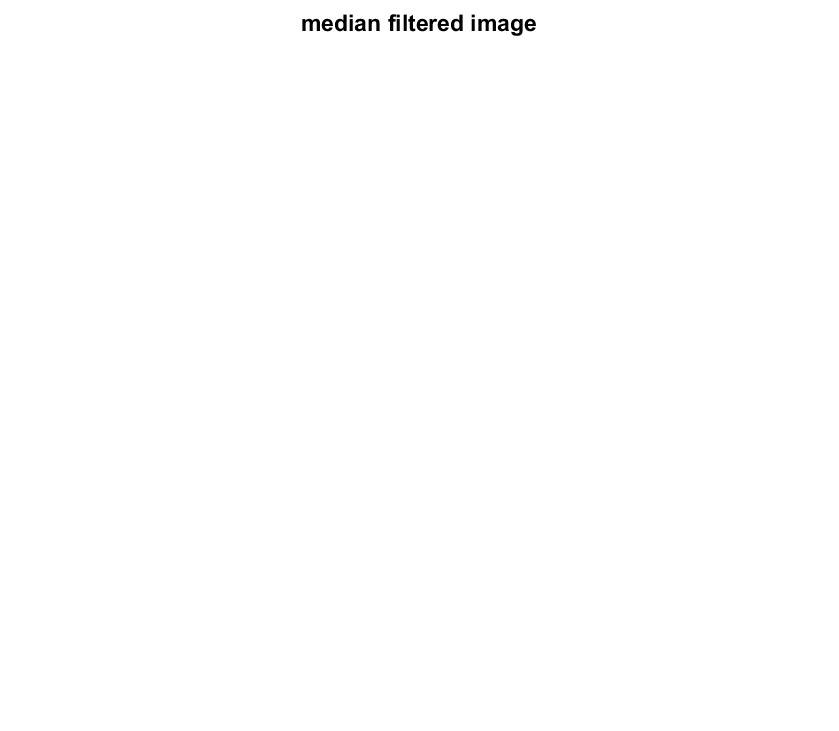


Figura 5: Implementação da filtragem median 2D por kernel 11x11.

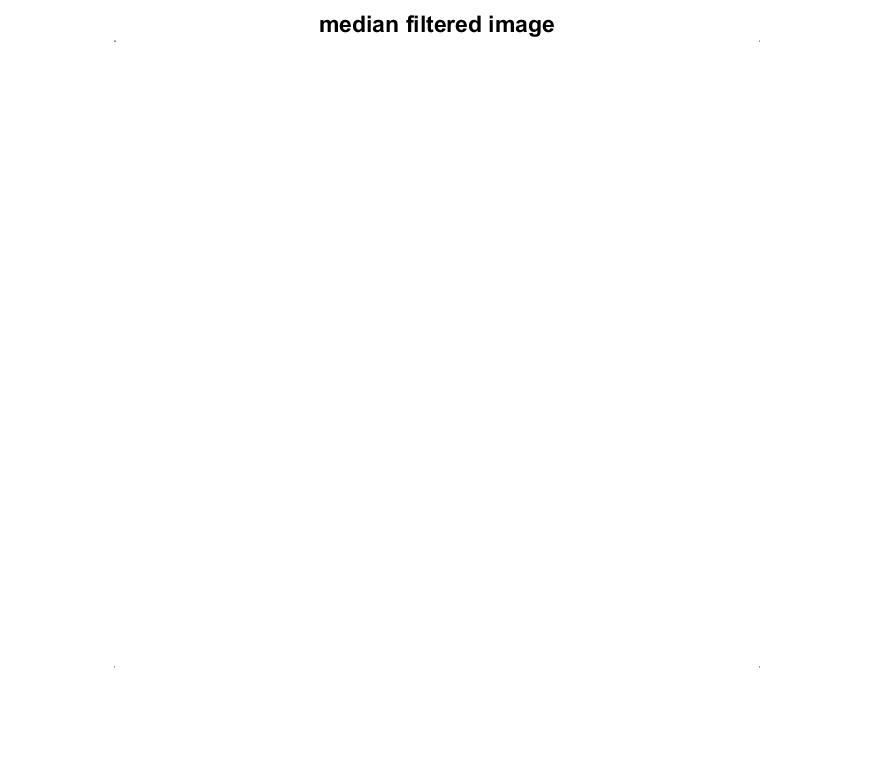


Figura 6: Implementação da filtragem median 2D por kernel 21x21.

Podemos observer na aplicação do filtro que o pixel central da imagem MxN pelo valor de cinza média da janela foram ordenados. Foi preservado, sem borrar, os contornos de regiões maiores que a do kernel utilizado e excluiu a descontinuidade que são menores que o tamanho da imagem.

Os elementos da mascara foram positivos, de modo que a saída é igual à entrada em regiões de constant intensidade. A quantidade de suavização e remoção de ruído é proporcional à dimensão do kernel.

**3.2 Função medianSFilter.m**

A função medianSFilter.m implementa um filtro de mediana aplicado a uma imagem com *padding* por replicação. O *padding* é feito automaticamente no corpo da função conforme solicitado no document de requisites do projeto.Como mostrado a seguir:

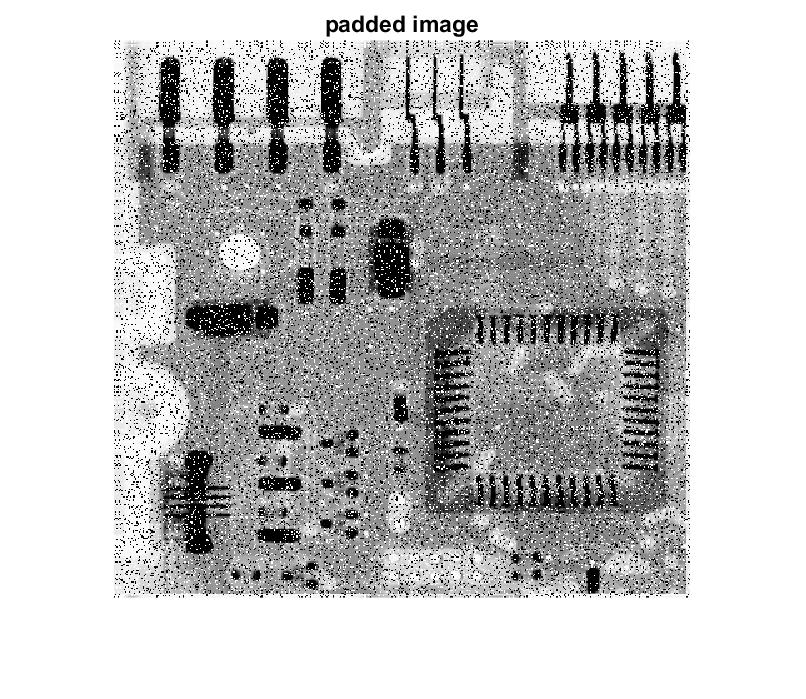


Figura 7: Imagem com padding de replicação.

Na imagem da figura 2 pode-se notar a filtragem espacial em 2D utilizando filtro de média. Sendo observando a implementação do padding da imagem por técnica replicate.

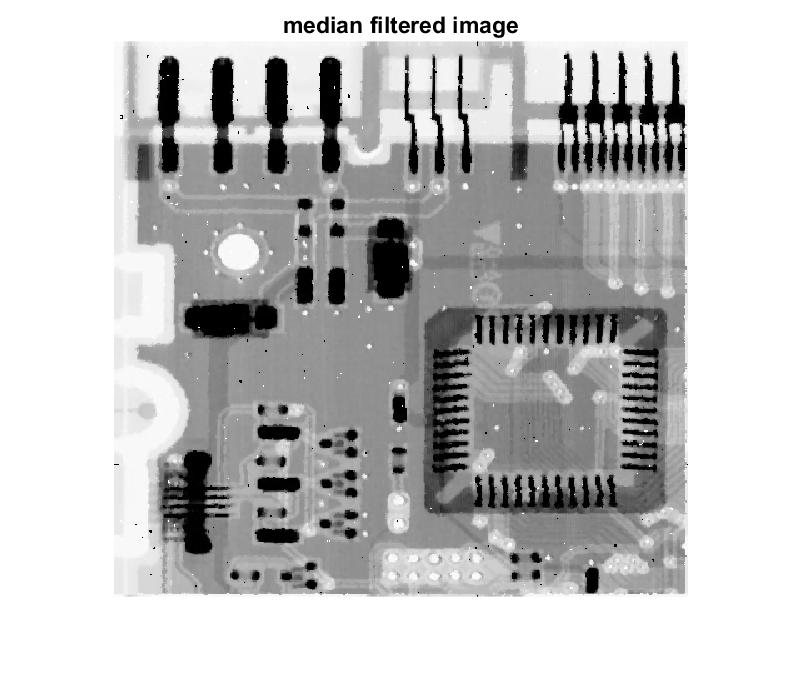


Figura 8 : Imagem com função medianSFilter.m aplicada

Nota-se na imagem da Figura 2, de um conjunto de valores MxN é tal que a metade dos valores são abaixo ou igual ao filtro de média, e metade são acima ou igual. Para a realização dessa filtragem num ponto da imagem, primeiro ordenamos os valores dos pixels na vizinhança,determinamos a média através da função medianSFilter.m, e atribuímos aquele valor ao pixel correspondente na imagem filtrada. Nessa aplicação observamos a suavização do ruído do tipo impulso, Transformada M, com as mesmas dimensões de MxN definidas para cada índices (i,j), o element M(i,j) da matriz transformada Aij.

1. **Referências**

Digital Image Processing Using MATLAB 2008 – R C Gonzales & R. E Woods, 2nd edition Prentice Hall