



UNIVERSIDADE FEDERAL DO MARANHÃO  
COORDENAÇÃO DO CURSO DE ENGENHARIA DA COMPUTAÇÃO

**Processamento de Imagens - Atividade 3<sup>a</sup> Nota**

**Aluno**

Claudio Henrique Velozo Alexandre  
Engenharia da Computação

**Professor**

Prof. Dr. MARCUS VINICIUS DE SOUSA LOPES  
Coordenação do Curso de Engenharia da Computação

São Luís, 2023

# 1 Tarefa 4

Para a tarefa 4 usou-se o filtro Notch, que são filtros capazes de rejeitar uma faixa bastante estreita de frequências, atuando quase que exclusivamente na frequência selecionada e bem pouco nas outras ao redor. Sua utilização é recomendada quando o sinal a ser atenuado é bem definido. Pelo fato de atuar em faixas reduzidas de frequências, filtros notch interferem pouco na qualidade do sinal.

A figura 1 que é a imagem a ser filtrada e a figura 2 mostra o espectro de frequência obtido ao aplicar a transformado de fourirer na imagem.

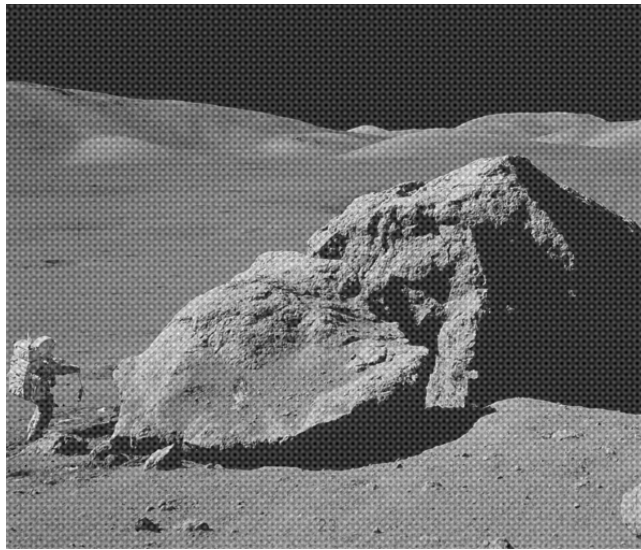


Figura 1: Imagem original



Figura 2: Espectro de Frequência da imagem original

Ao perceber e localizar os picos de frequência, implementou-se uma função para rejeitar

essa faixa de frequência que pode ser vista na figura 3 a seguir:

```
1 import numpy as np
2
3
4 def filtro_rejeita_notch(shape, d0=9, u_k=0, v_k=0):
5     (M, N) = shape
6
7     H_0_u = np.repeat(np.arange(M), N).reshape((M, N))
8     H_0_v = np.repeat(np.arange(N), M).reshape((N, M)).transpose()
9
10    D_uv = np.sqrt((H_0_u - M / 2 + u_k) ** 2 + (H_0_v - N / 2 + v_k) ** 2)
11    D_muv = np.sqrt((H_0_u - M / 2 - u_k) ** 2 + (H_0_v - N / 2 - v_k) ** 2)
12
13    seletor_1 = D_uv ≤ d0
14    seletor_2 = D_muv ≤ d0
15
16    seletor = np.logical_or(seletor_1, seletor_2)
17
18    H = np.ones((M, N))
19    H[seletor] = 0
20
21    return H
22
```

Figura 3: Filtro Notch

O espectro de frequência com o filtro notch pode ser visto na figura 4 abaixo. Note que tentou-se esconder os picos de frequência utilizando círculos pretos:

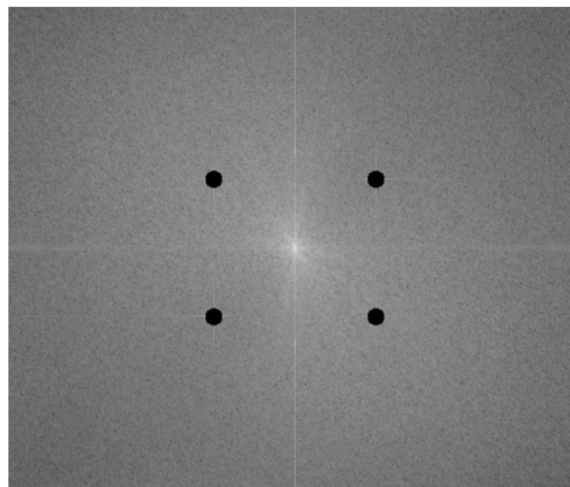


Figura 4: Espectro do Filtro Notch

A figura 5 mostra o resultado da filtragem. A filtragem não ficou 100% mas é possível perceber a redução do ruído em comparação a imagem original.



Figura 5: Imagem resultante da filtragem

## 2 Tarefa 5

Para a tarefa 5 o filtro Notch também foi utilizado.

Podemos observar os ruídos da imagem original e seu espectro de frequência nas figuras 6 e 7 respectivamente.



Figura 6: Imagem original

A mesma função da figura 3 foi utilizada, mudando apenas as coordenadas dos pixels para os picos de frequência.

O filtro notch pode ser visto na figura 8 e o resultado está na figura 9.

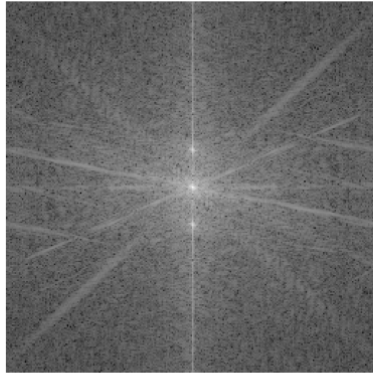


Figura 7: Espectro de Frequência da imagem original

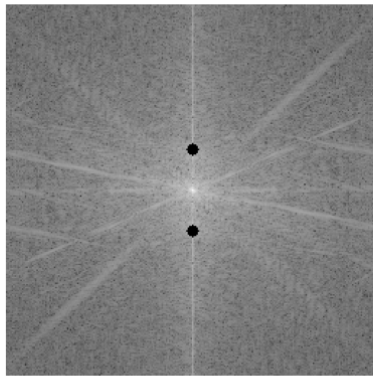


Figura 8: Espectro do Filtro Notch



Figura 9: Imagem resultante da filtragem

### 3 Implementação

Os código completos podem ser obtidos no meu repositório no github.

link: <<https://github.com/cHenrique0/processamento-imagens-ufma/tree/main/provas/P3>>