

DIM0888 - Lista de Exercícios 1

Danilo Rodrigo Cavalcante Bandeira (20181006147)

15/05/2018

- 1 Descreva com suas palavras, o que são amostragem e quantização. Dada a função $f(x) = x^2 - 3.5x + 1$, gere o código binário que representa o sinal digitalizado usando-se uma taxa de amostragem de 0.5 unidades no intervalo $[0, 5]$ e 16 níveis de cinza.**

Amostragem (sampling) e Quantização (quantization) são dois processos usados em conjuntos para converter um sinal contínuo em um sinal discreto, no caso das imagens por exemplo, a mesma é adquirida de forma analógica por um aparelho e convertida em uma imagem digital, passando de um sinal contínuo para um discreto. A amostragem diz respeito ao eixo “x” do sinal, em outras palavras, a amostragem irá dividir o sinal em um intervalo finito e discreto, para isso uma taxa de amostragem é usada com o propósito de determinar a distância entre cada elemento do intervalo discretizado. A quantização diz respeito ao eixo “y” do sinal, em outras palavras, o processo de quantização irá determinar um valor de intensidade do sinal para cada elemento do intervalo obtido pela amostragem. É importante ressaltar que quanto menor for a taxa de amostragem maior será a fidelidade do sinal discretizado com o sinal original.

Para gerar o código binário da função $f(x) = x^2 - 3.5x + 1$, o primeiro passo é realizar o processo de amostragem e quantização para o intervalo dado ($[0, 5]$) obedecendo a taxa de amostragem de 0.5. A realização desse processo nos dá os valores x que iremos usar no processo de quantização. A Tabela 1 a seguir nos mostra os valores obtidos pelo processo de amostragem na coluna x , a função para cada valor do intervalo discretizado na coluna $f(x)$ (quantização) e o valor referente a cada elemento de x na coluna y .

Tabela 1: Valores de $f(x)$

| x | Substituição em $f(x)$ | $f(x)$ |
|-----|--------------------------------------|--------|
| 0 | $f(0) = 0^2 - 3.5 \cdot 0 + 1$ | 1 |
| 0.5 | $f(0.5) = 0.5^2 - 3.5 \cdot 0.5 + 1$ | -0.5 |
| 1 | $f(1) = 1^2 - 3.5 \cdot 1 + 1$ | -1.5 |
| 1.5 | $f(1.5) = 1.5^2 - 3.5 \cdot 1.5 + 1$ | -2 |
| 2 | $f(2) = 2^2 - 3.5 \cdot 2 + 1$ | -2 |
| 2.5 | $f(2.5) = 2.5^2 - 3.5 \cdot 2.5 + 1$ | -1.5 |
| 3 | $f(3) = 3^2 - 3.5 \cdot 3 + 1$ | -0.5 |
| 3.5 | $f(3.5) = 3.5^2 - 3.5 \cdot 3.5 + 1$ | -1 |
| 4 | $f(4) = 4^2 - 3.5 \cdot 4 + 1$ | 3 |
| 4.5 | $f(4.5) = 4.5^2 - 3.5 \cdot 4.5 + 1$ | 5.5 |
| 5 | $f(5) = 5^2 - 3.5 \cdot 5 + 1$ | 8.5 |

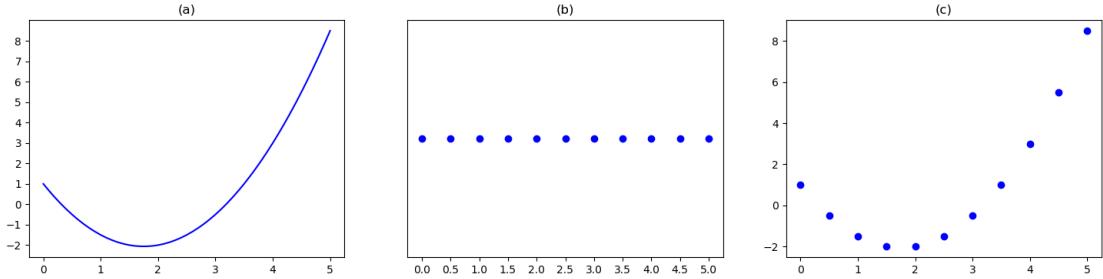


Figura 1: Processo de amostragem e quantização da função

A Figura 1 acima demonstra o processo de amostragem e quantização dada, a imagem (a) mostra função contínua, a imagem (b) o número de pontos usados para representar o sinal (definido pela taxa de amostragem) e a imagem (c) os pontos da imagem anterior após a quantização. Para mapear os valores discretizados pelo processo anterior de amostragem e quantização para 16 níveis de tons de cinza ($L=16$), foi aplicada uma técnica de normalização de Mínimo e Máximo dada pela Equação 1, onde x é o valor que se deseja normalizar.

$$\text{normalized} = \frac{(newMax - newMin)(x - oldMin)}{(oldMax - oldMin)} + newMin \quad (1)$$

A Tabela a seguir apresenta os valores após a normalização, arredondados (para facilitar a representação binária) e seu valor em números binários depois de normalizados. Ficando assim com o seguinte código binário final: 0100 0010 0001 0000 0000 0001 0010 0100 0111 1011 1111.

Tabela 2: Valores normalizados e convertidos para binário

| x | Normalizado | Aproximado | Binário (4 bits) |
|------|-------------|------------|------------------|
| 1 | 4.28571429 | 4 | 0100 |
| -0.5 | 2.14285714 | 2 | 0010 |
| -1.5 | 0.71428571 | 1 | 0001 |
| -2 | 0 | 0 | 0000 |
| -2 | 0 | 0 | 0000 |
| -1.5 | 0.71428571 | 1 | 0001 |
| -0.5 | 2.14285714 | 2 | 0010 |
| 1 | 4.28571429 | 4 | 0100 |
| 3 | 7.14285714 | 7 | 0111 |
| 5.5 | 10.71428571 | 11 | 1011 |
| 8.5 | 15 | 15 | 1111 |

2 Como a operação de convolução é implementada no domínio espacial? Como a operação de convolução é implementada no domínio da frequência?

A operação de convolução é implementada no domínio espacial passando um filtro de tamanho $M \times N$, contendo coeficientes, sobre uma imagem, porém, antes da aplicação o mesmo é rotacionado sobre o seu vértice em 180° . Para cada deslocamento do filtro o pixel da imagem correspondente a posição central do filtro receberá o valor da soma da multiplicação “pixel a pixel” dos elementos do filtro pelos seus correspondentes na imagem. Os valores dos coeficientes serão determinados de acordo com a funcionalidade que se deseja

alcançar. Caso o filtro seja simétrico, as operações de convolução e correlação serão exatamente iguais. Abaixo se encontra a equação da convolução do domínio espacial.

$$w(x, y) \star f(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x - s, y - t) \quad (2)$$

Em se falando do domínio da frequência a convolução funciona de forma similar ao domínio espacial, porém, por se tratar de um sinal contínuo, ao invés de realizar um somatório dos produtos, um integração é feita rotacionando uma das funções (assim como no domínio espacial) e passando-a sobre a outra. Comparando a Equação 2 com a Equação 3 abaixo, podemos notar que a diferença entre ambas está no fato de uma somar os resultados dos produtos enquanto a outra integra.

$$f(t) \star h(t) = \int_{-\infty}^{\infty} f(\tau) h(t - \tau) d\tau \quad (3)$$

3 Defina a operação de correlação. Qual sua utilidade?

A correlação consiste em aplicar um filtro de tamanho $M \times N$ sobre uma imagem, onde cada elemento do filtro possui um coeficiente. A aplicação se dá somando os valores da multiplicação entre cada elemento do filtro e o pixel da imagem na posição correspondente, o resultado desta soma de produtos é atribuído ao pixel que se encontra na posição correspondente ao elemento central do filtro. Para facilitar a sua aplicação é muito comum utilizar filtros de tamanho ímpares e adicionar zeros ao redor da imagem.. A correlação é definida pela Equação 4 abaixo, podemos notar que a única diferença entre a correlação e convolução domínio espacial esta no sinal negativo da função correspondente ao filtro, que significa exatamente a rotação de 180° .

$$w(x, y) \star f(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t) \quad (4)$$

A principal utilidade da correlação é fazer uma filtragem em uma imagem usando como base a vizinhança entre os pixels para diversos propósitos. Os mais comuns são: atenuar ou suavizar características, comparar e detectar padrões e diminuir ou eliminar ruídos. O que determina qual funcionalidade uma filtragem usando correlação irá exercer são os coeficientes do filtro que será multiplicado pela imagem.

4 Dada a propriedade da translação das DFTS (ver abaixo), mostre que $f(x, y)(-1)^{x+y} \Leftrightarrow F(u - M/2, v - N/2)$

$$f(x, y)e^{j2\pi(u_0x/M+v_0y/N)} \Leftrightarrow F(u - u_0, v - v_0) \quad (5)$$

$$f(x - x_0, y - y_0) \Leftrightarrow F(u, v)e^{-j2\pi(u_0x/M+v_0y/N)} \quad (6)$$

Dada a propriedade da Equação 5, o primeiro passo é escolher os valores para u_0 e v_0 como sendo $M/2$ e $N/2$ respectivamente, para poder mover a função para o centro do plano, ficando com a seguinte equação:

$$f(x, y)e^{j2\pi(\frac{M}{2}\frac{x}{M}+\frac{N}{2}\frac{y}{N})} \Leftrightarrow F(u - M/2, v - N/2) \quad (7)$$

Simplificando a Equação 7 chegamos em:

$$f(x, y)e^{j\pi(x+y)} \Leftrightarrow F(u - M/2, v - N/2) \quad (8)$$

Usando a Identidade de Euler expressa pela equação $e^{ix} = \cos x + i \sin x$, temos que $e^{j\pi} = -1$, como demostrado abaixo:

$$\begin{aligned} e^{j\pi} &= \cos \pi + j \sin \pi \\ e^{j\pi} &= -1 + j \cdot 0 \\ e^{j\pi} &= -1 \end{aligned} \tag{9}$$

Aplicando em $e^{j\pi(x+y)}$, encontrado na Equação 8, o resultado encontrado na Equação 9, chegamos na Equação 10 abaixo, que é exatamente o que se deseja demonstrar.

$$f(x, y)(-1)^{x+y} \Leftrightarrow F(u - M/2, v - N/2) \tag{10}$$

5 Implemente a técnica de equalização de histogramas e a aplique nas imagens do conjunto de testes que possuem baixo contraste. Comente sobre os resultados.

Após a aplicação da equalização de histograma nas imagens de baixo contraste a melhora foi notável. As transições passaram a ser mais aparentes e possibilidade de visualizar mais detalhes foi claramente aumentada. Entretanto, também é possível ver pequenas falhas, principalmente das regiões de mais baixas frequências, mas que em um contexto geral não diminuem a qualidade dos resultados obtidos.

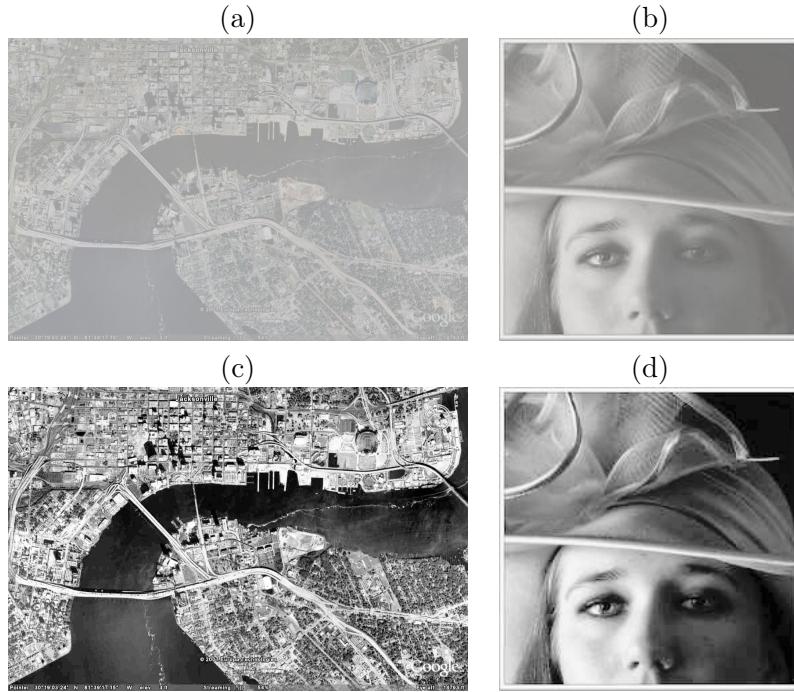


Figura 2: (a) e (b) imagens originais com baixo contraste (c) e (d) Imagens depois da equalização de histograma

Abaixo pode-se observar os histogramas das imagens da Figura 3 antes e depois da equalização.

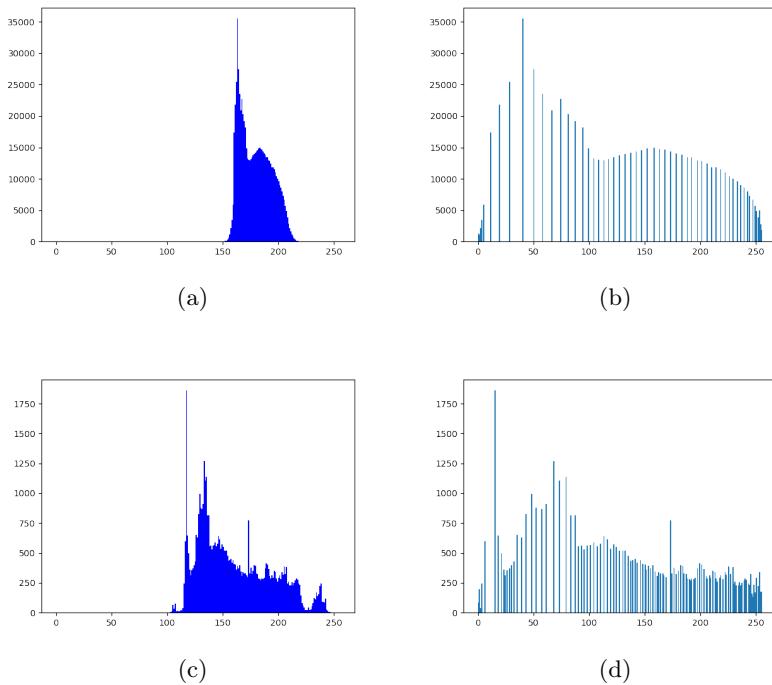


Figura 3: (a) e (b) histogramas da imagem “Google JAX low contrast.jpg” antes e depois da equalização, respectivamente (c) e (d) histogramas da imagem “woman low contrast.jpg” antes e depois da equalização, respectivamente

6 Implemente o filtro da mediana e o da média contra-harmônica e os aplique nas imagens do conjunto de testes que seriam melhoradas pelo seu uso (todas as imagens afetadas pro ruído sal e pimenta), comparando seus resultados.

Em todas as imagens testadas o filtro da mediana apresentou um resultado superior ao da contra-harmônica, devido ao fato deste segunda só eliminar um ruído por vez (ou sal ou pimenta) enquanto o da mediana elimina ambos. A Figura 4 abaixo mostra o resultado do filtro da mediana com tamanho 3 ($L = 3$) em algumas das imagens de teste, é notória a melhora da imagem até na imagem do pulmão com 50% de ruído. A contra partida deste filtro está no fato de haver um aumento do borramento ao passo que se aumenta o tamanho da máscara.

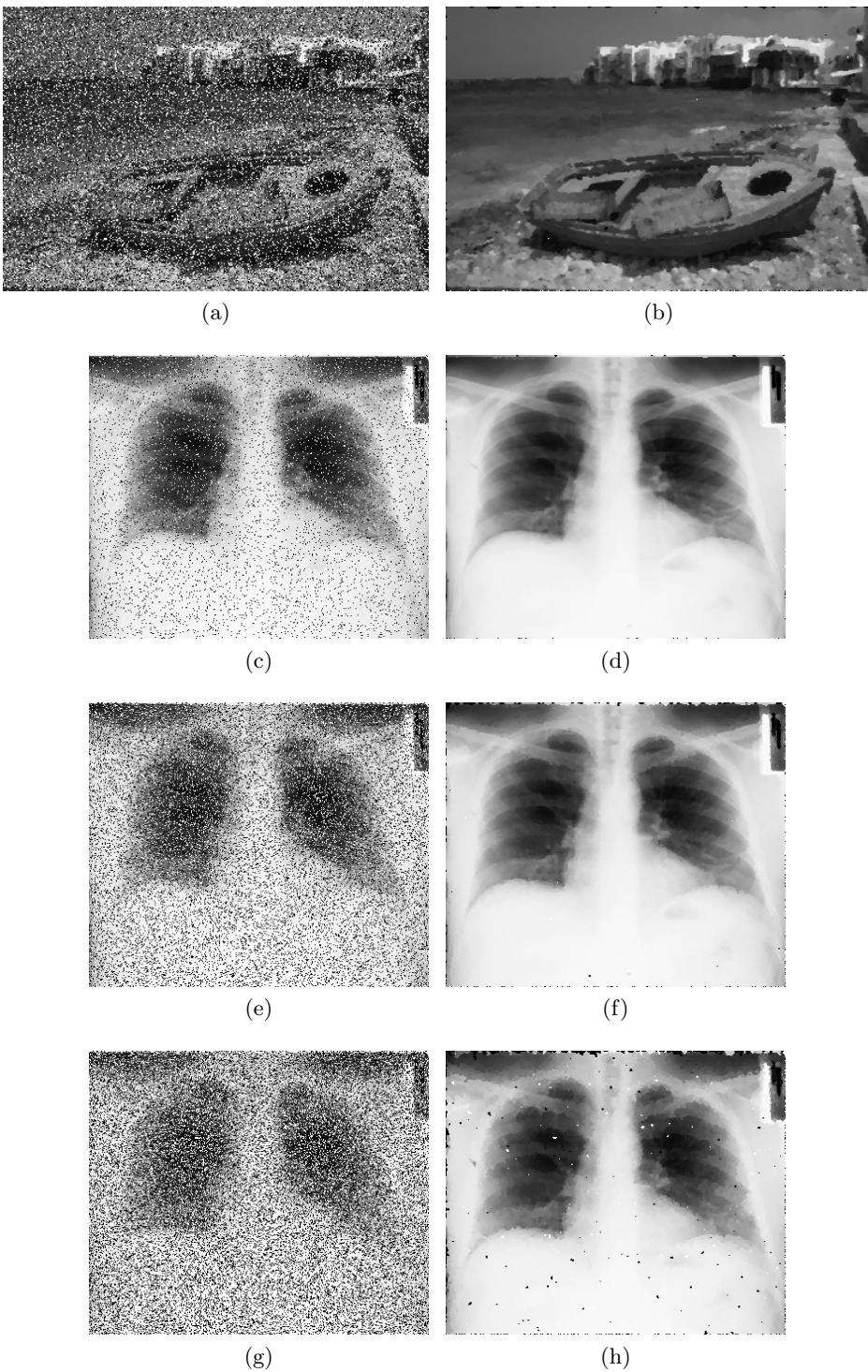


Figura 4: (a), (c), (e) e (g) imagens originais, (b), (d), (f) e (h) imagens após i filtro da maediana de tamanho 3×3

O filtro da contra-harmônica alcançou uma redução não muito satisfatória dos ruídos nem de pimenta com $q = 1,5$ nem de sal ($q = -1,5$), em alguns casos pode-se notar um aumento do tamanho do ruído. A Figura 5 mostra uma das imagens de teste com o $q = [1, -1, 0]$.

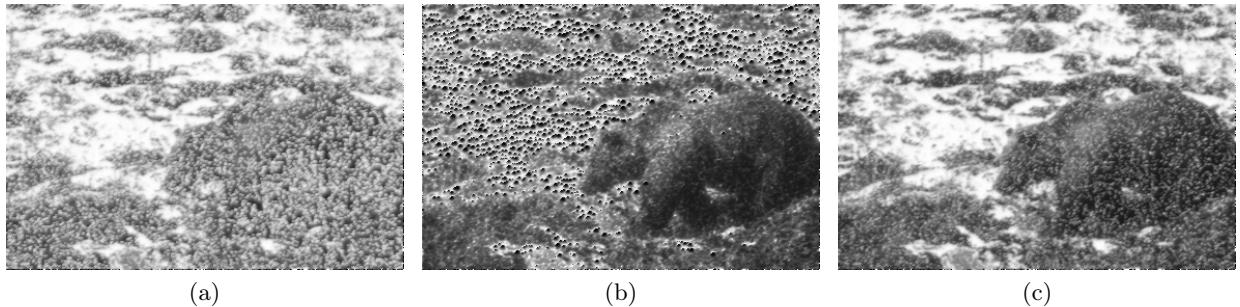


Figura 5: (a) imagem com $q = 1$ (b) imagem com $q = -1$ (c) imagem com $q = 0$.

7 Implemente o filtro high-boost e o aplique nas imagens do conjunto de testes que possuem baixo contraste, antes e depois da aplicação da equalização de histogramas (Google JAX low contrast.jpg e woman low contrast.jpg). Comente sobre os resultados.

O filtro hight-boost foi implementado usando como base o livro do Woods e Gonzales, para isso o algoritmo segue os seguintes passos:

1. aplicação do filtro da média (borramento), de tamanho L , sobre a imagem original.
2. Subtração do resultado do passo anterior pela imagem original (chamado de máscara).
3. Multiplicação da máscara por K , onde o mesmo é um número inteiro que indicará o quanto se quer ressaltar as altas frequências.
4. Adição da máscara multiplicada por K à imagem original.

A Figura 6 a seguir apresenta um exemplo dos passos apresentados para a imagem “Google JAX low contrast.jpg” com $K = 3$ e $L = 2$.

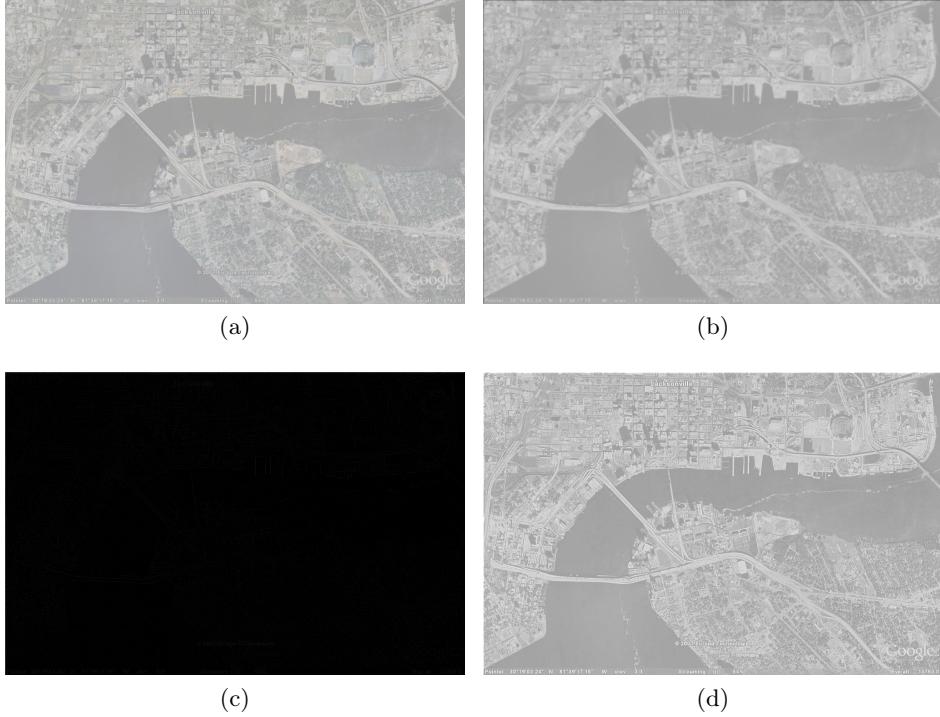


Figura 6: (a) imagem original (b) imagem após o borramento (c) máscara gerada (d) imagem com high-boost

Após a aplicação do high-boost foi notada uma aparente aumento da altas frequências nas imagens antes e após a equalização de histograma. Nas imagens não equalizadas, a aplicação do filtro apresentou uma ótima melhora do contraste, mantendo uma aparência bem natural. O aumento do K gerou um aumento no brilho da imagem, enquanto o aumento do L gerou um aumento no contraste. Entretanto, com um valor elevado K gera o aparecimento de pequenos quadrados em áreas de baixa frequência.

Já nas imagens equalizadas o aparecimento de pequenos quadrados em áreas de baixa frequência ocorre com um valor de K relativamente baixo, mas mantendo um aumento do brilho de acordo com o aumento de K . Em relação ao L o seu aumento continuou a apresentar uma relação diretamente proporcional com o contraste da imagem.

Em geral, ambas as imagens, equalizadas ou não, mostram um destaque em áreas de alta frequência. Entretanto, as imagens equalizadas apresentam uma maior sensibilidade quanto a variação do K e do L , além das imagens não equalizadas obterem um efeito mais natural após o high-boost.

8 Implemente o filtro rejeita-banda notch ideal no domínio da frequência e aplique nas imagens do conjunto de testes que seriam melhoradas pelo seu uso (`mit noise periodic.png` e `pompei periodic noise.png`). Comente sobre os resultados.

Para a implementação do filtro notch ideal, o espectro de Fourier é apresentado ao usuário e mesmo deve selecionar a área que se deseja manter com base no ponto central e final da seleção é obtido o raio dessa área. Com posse dessa informação uma máscara de mesmo tamanho do espectro é criada contendo valores zero para o espaço não selecionado pelo usuário e 1 para o espaço selecionado. A imagem e a máscara são multiplicadas e a IDFT do resultado é calculada, obtendo assim a imagem final.

A aplicação do filtro ideal elimina boa parte do ruído periódico, dependendo da área selecionada pelo usuário. Entretanto, alguma informação é perdida durante o processo, resultando em uma imagem levemente

borrada. Na Figura 7 a seguir pode-se observar o processo assim como o resultado da aplicação do filtro para a imagem “pompei periodic noise.png”. A Equação 11 expressão como a máscara foi gerada.

$$D(u, v) = [(u - P/2)^2 + (v - Q/2)^2]^{1/2} \quad (11)$$

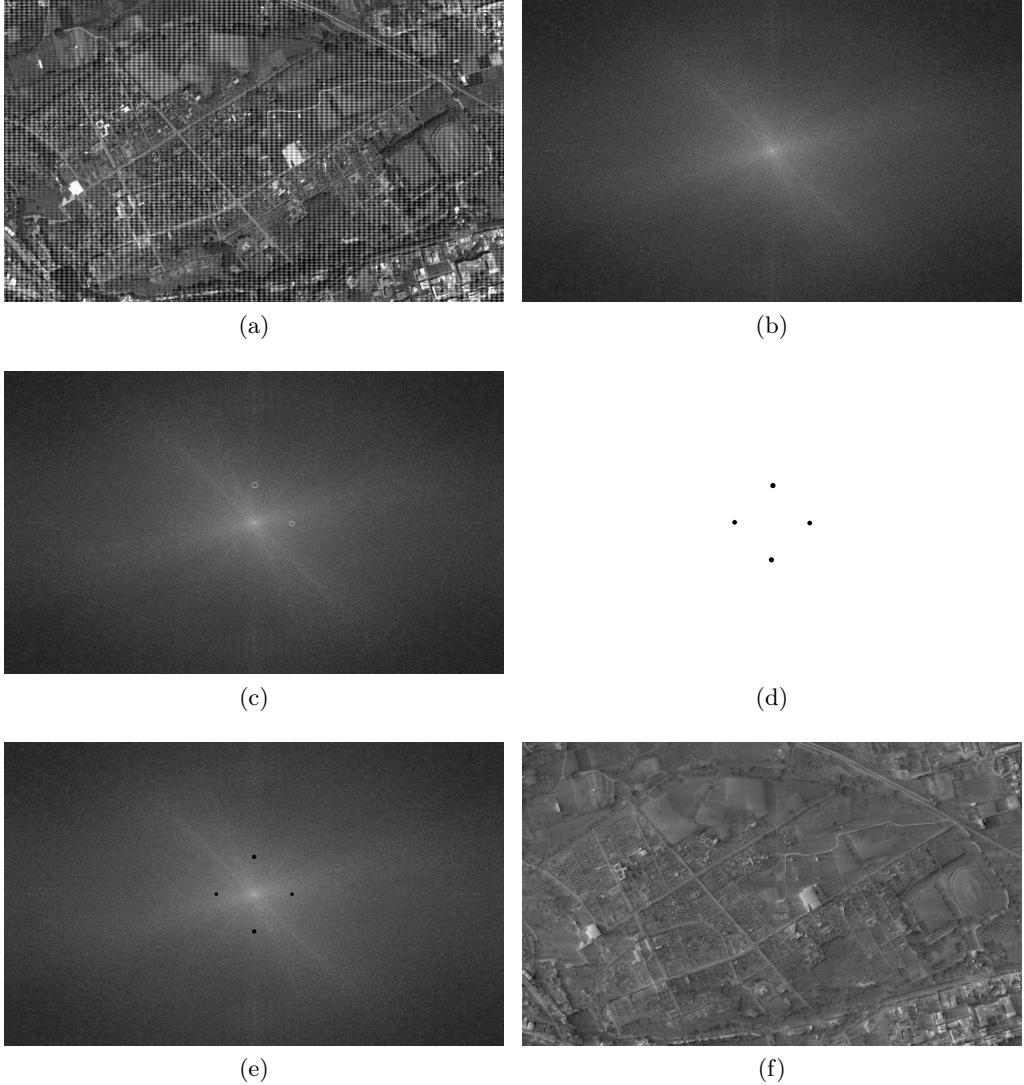


Figura 7: (a) imagem original (b) espectro de Furier (c) área selecionada pelo usuário (d) máscara gerada a partir da seleção (e) espectro após convolução com a máscara (f) resultado após aplicação da IDFT na imagem anterior.

9 Implemente o filtro rejeita-banda de Butterworth no domínio da frequência e aplique nas imagens do conjunto de testes que seriam melhoradas pelo seu uso (Man on the moon - noisy.png). Comente sobre os resultados.

A implementação do filtro rejeita-banda de Butterworth se da de forma parecida com o filtro ideal, porém, as variáveis são controladas no código. A Equação abaixo expressa a fórmula usada, as variáveis

D_0 e n controlam o raio e o gradiente do filtro respectivamente. Após a aplicação do filtro na imagem a mesma obteve uma relativa melhora eliminando boa parte do ruído, porém com um aumento considerável do borramento. Com $n = 1$ a área em que o degradê aparece é ampla, o que ocasiona uma não eliminação por completo do ruído. A Figura apresenta o passo a passo do processo.

$$H(u, v) = \frac{1}{1 + [D(u, v)/D_0]^{2n}} \quad (12)$$

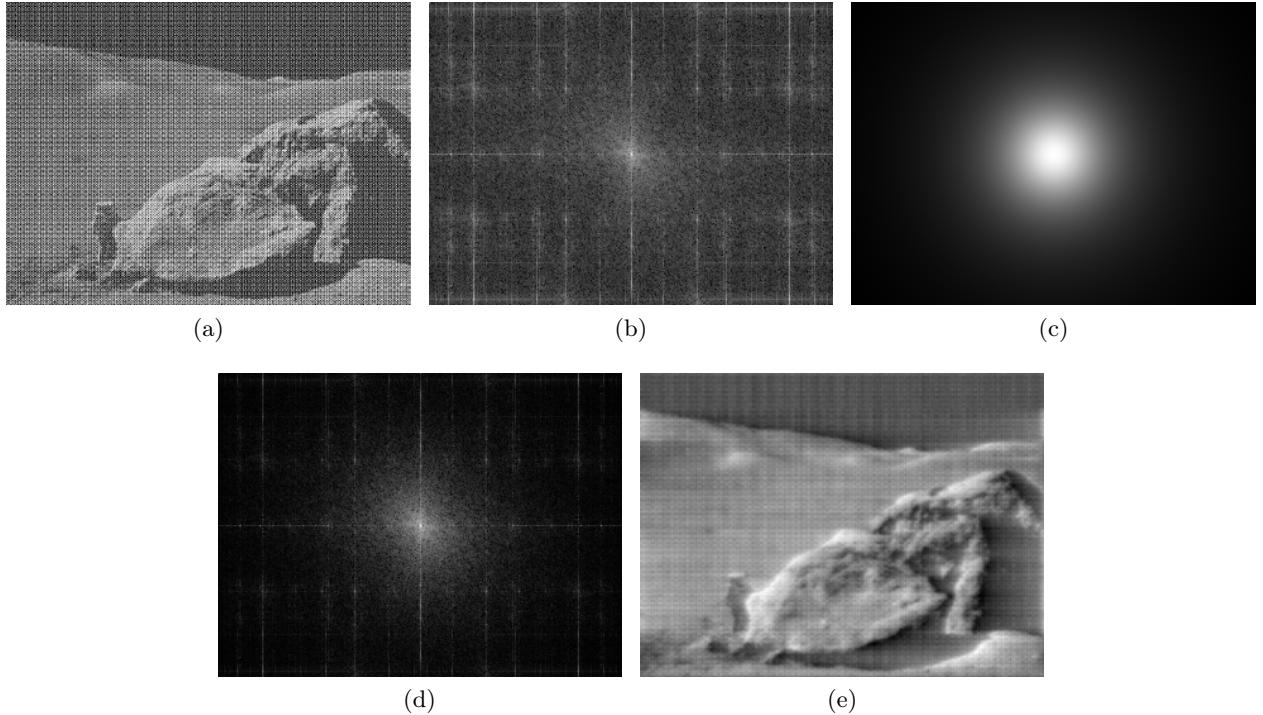


Figura 8: (a) imagem original (b) espectro de Furier (c) máscara gerada a partir da seleção (d) espectro após convolução com a máscara (e) resultado após aplicação da IDFT na imagem anterior.

A Figura 12 acima apresenta o passo a passo do processo de aplicação do filtro, as imagens foram obtidas usando $n = 1; D_0 = 25$. A resultado não tão satisfatório pode ser explicado observado a máscara, pois devido ao efeito do Butterworth ao se afastar do cetro a máscara vai suavizando de forma gradativa as frequências, deste modo, muito do ruído ainda é mantido apesar de suavizado.

10 Utilize o filtro de Wiener para restaurar a imagem de teste que apresenta borramento linear (lighthouse blurred.png). Calcule o RMSE usando a imagem original. Comente sobre os resultados.

Para aplicação do filtro de Wiener para a recuperação de uma imagem borrada a partir da imagem original, os seguintes passos foram seguidos:

1. Converter as imagens para *grayscale*;
2. Calcular a DFT da imagem original e da corrompida;
3. Estimar a deterioração usando os dois espectros;

4. Gerar uma máscara de deterioração a partir do passo anterior;
5. Aplicar a Equação 13 com G sendo a imagem original, H sendo a máscara e K sendo uma constante a se estimar; e
6. Calcular a IDFT do resultado do passo anterior.

$$\hat{F}(u, v) = \left[\frac{1}{H(u, v)} \frac{|H(u, v)^2|}{|H(u, v)^2| + K} \right] G(u, v) \quad (13)$$

Para estimar a degradação da imagem original e gerar a máscara, optei por dividir “ponto a ponto” a imagem original pela borrada, fazendo os testes com $K = [0, 1, 2]$. Com o $K = 0$ a recuperação da imagem borrada alcançou níveis ótimos, enquanto ao passo em que o K aumentava, um efeito semelhante ao sépia era notado na imagem. A escolha de dividir os espectros de forma completa, assim como de converter para *grayscale*, foi tomada após uma grande dificuldade em estimar a degradação em áreas isoladas. A Figura 9 mostra o processo da aplicação do filtro e a Figura 10 mostra o resultado para diferentes valores de K .

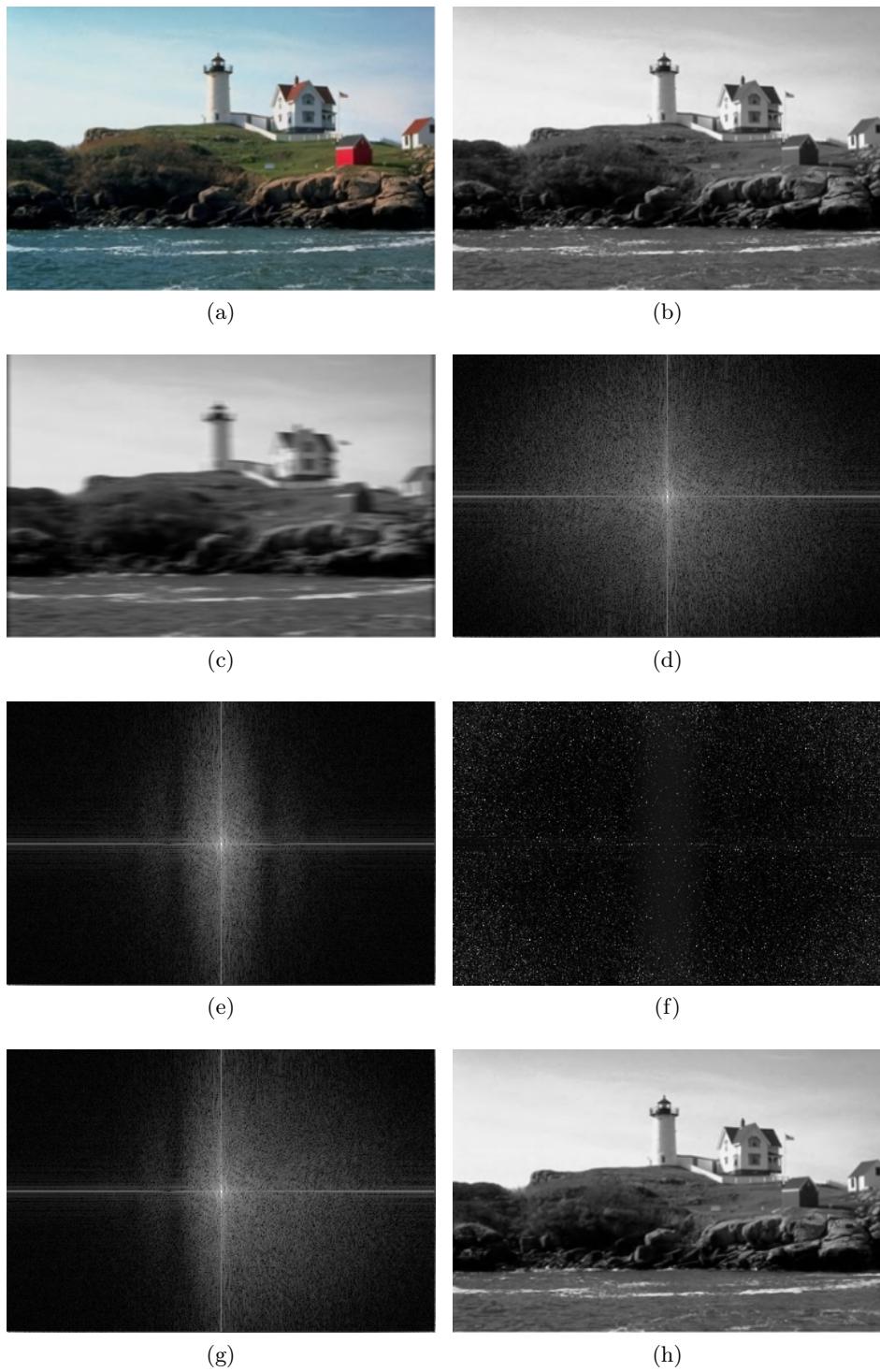


Figura 9: (a) imagem original (b) imagem em *grayscale* (c) imagem borrada em *grayscale* (d) espectro de (b) (e) espectro de (c) (f) máscara (g) espectro após a aplicação da máscara (g) imagem após a aplicação do filtro de Wiener.

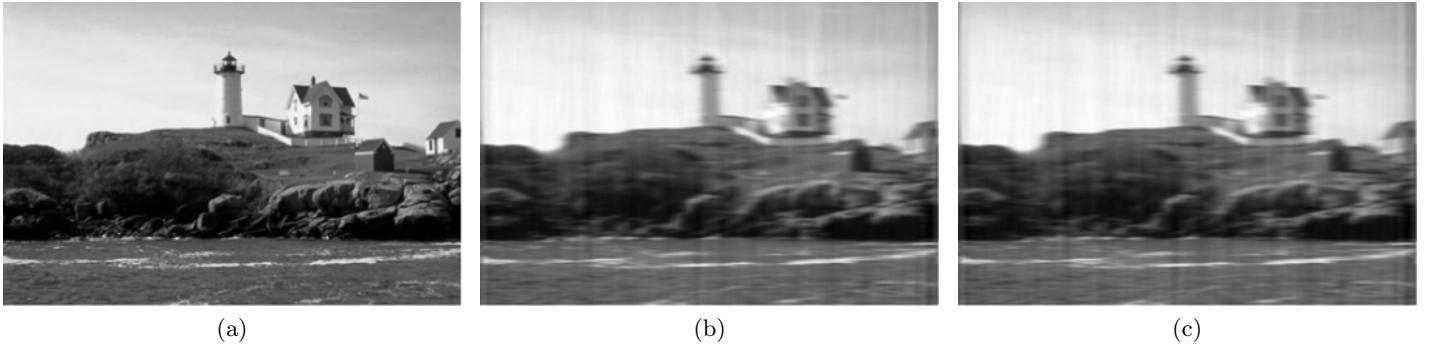


Figura 10: (a) Filtro de Wiener com $k = 0$ (b) Filtro de Wiener com $k = 1$ (c) Filtro de Wiener com $k = 2$.

11 Sobre o código

A pasta submetida está organizada de acordo com a Figura 11 contendo uma pasta com a saída dos códigos executados anteriormente (output_original), uma pasta para salvar os resultados dos testes realizados pelo professor (output), uma pasta com as imagens renomeadas para facilitar a inserção no L^AT_EX(images), uma pasta contendo relatório (paper) e os códigos de cada questão (question5.cpp à question10.cpp). Alguns código usam um *array* com as imagens que serão usadas bastando apenas comentar/descomentar quais desejam aplicar os filtros, assim como quaisquer alterações de parâmetros serão ser realizadas no código (os locais estão comentados), já algumas questões usam as imagens de origem *hard coded*.

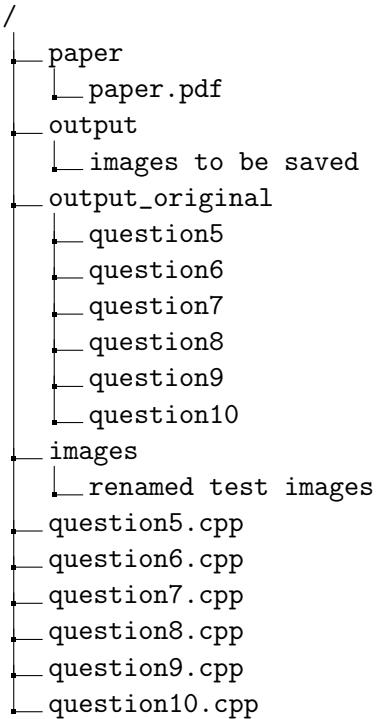


Figura 11: Organização do diretório submetido.

Durante a realização da lista foram testadas as versões 2.4 e 3.4 do OpenCV, funcionando em ambas.

Para a execução de um arquivo “.cpp” foi utilizada a seguinte linha de comando: “g++ file.cpp -std=c++11 -o file ‘pkg-config --cflags --libs opencv’ ”.