

Trabalho 03 de Introdução ao Processamento de Imagem Digital

MARIANNA DE PINHO SEVERO

RA: 264960

I. INTRODUÇÃO

O processamento de imagens é uma importante atividade desempenhada dentro da área de Visão Computacional, sendo responsável pela captura, representação e transformação de imagens. Ele permite a extração de informações e a transformação de certas características de imagens, de maneira a facilitar tanto a percepção humana como a interpretação realizada por computadores [1].

Com respeito à transformação de imagens, diversas técnicas podem ser utilizadas, de acordo tanto com as necessidades do domínio de aplicação em que as imagens serão empregadas, como com os objetivos das transformações.

Neste trabalho, realizamos a filtragem de imagens no domínio de frequência por meio de três tipos de filtros ideais, pertencentes às classes passa-baixa, passa-alta e passa-faixa. Além disso, empregamos uma técnica de compressão de imagens por meio do anulamento de uma determinada proporção de coeficientes do espectro de Fourier de uma imagem dada como entrada.

A filtragem é uma importante etapa da transformação de imagens, utilizada para diversas finalidades, como redução de ruídos, redução de detalhes irrelevantes e realçamento de bordas e de pequenas descontinuidades. Por sua vez, a compressão é uma importante etapa do processamento de imagens que permite que a informação importante presente em uma imagem seja representada por estruturas que utilizam menor quantidade de dados, viabilizando grande maioria das aplicações que observamos atualmente.

Dessa maneira, os resultados obtidos pela aplicação de filtros permite a utilização de processamento de imagens para diversas aplicações do mundo real, como inspeção industrial, imagiologia médica, direção automática, detecção de objetos, entre muitas outras. E aqueles obtidos pela compressão de imagens permite a transmissão mais veloz destas por meio de conexões na Internet, o melhor aproveitamento do armazenamento fornecido por dispositivos como cartões de memória e discos rígidos, a realização de atividades como sensoriamento remoto, vídeo conferências, e muitas outras [2].

Na Seção 2 são apresentadas as bibliotecas utilizadas neste trabalho; na Seção 3 são descritas a saída e entrada de dados; na Seção 4 apresenta-se as técnicas empregadas para a aplicação dos filtros e da compressão; na Seção 5 são apresentados os resultados e discussões; por fim, na Seção 6, apresenta-se a conclusão.

II. DEPENDÊNCIAS

Para a implementação dos algoritmos utilizados neste trabalho, as seguintes bibliotecas foram empregadas:

- numpy: usada para a manipulação dos *arrays* que representam as imagens e para as transformações de fourier.
- opencv: utilizada para a leitura e escrita de imagens e para as transformações de fourier.
- matplotlib: empregada para a apresentação das imagens.
- skimage: utilizada para a escrita de imagens.

III. ENTRADA E SAÍDA DE DADOS

As imagens utilizadas para os testes neste trabalho foram retiradas do [Site da Disciplina](#) e estão todas no formato PNG (*Portable Network Graphics*). Elas foram armazenadas em um diretório chamado **input_images**.

Para utilizá-las, elas foram carregadas utilizando o trecho de código 1, em que o primeiro parâmetro indica o caminho da imagem e o segundo informa o formato para o qual queremos carregá-las, neste caso, 0 indica que carregaremos uma imagem no formato monocromático. A função empregada retorna um *array* numpy em que cada elemento representa a intensidade de um pixel da imagem.

```
image = cv.imread('input_images/name_image.png', 0)
```

Código 1: Carregar uma imagem com OpenCV.

Por sua vez, as imagens geradas - que também são representadas por *arrays* numpy no programa - são armazenadas, também no formato PNG, em cinco diretórios, de acordo com o tipo de filtragem e de compressão. Os diretórios para as imagens filtradas são: **output_images/LPF** para os resultados dos filtros passa-baixa; **output_images/HPF**, para aqueles dos filtros passa-alta; e **output_images/BPF** para os resultados dos filtros passa-banda.

Já as imagens resultantes da compressão são armazenadas nos diretórios: **output_images/compressed/threshold**, para as compressões feitas utilizando-se um valor limitante; e **output_images/compressed/percent** para aquelas em que se zerou uma porcentagem da quantidade de coeficientes, do menor para o maior valor de magnitude dos coeficientes. Para salvar as imagens, utilizamos o trecho de código 2, em que o primeiro argumento é o caminho onde salvaremos a imagem, sendo *subdir* um dos subdiretórios citados, e o segundo é a estrutura de dados que a representa.

```
cv.imwrite('output_images/subdir/name_image.png',  
          image)
```

Código 2: Salvar uma imagem com OpenCV.

Os algoritmos implementados neste trabalho foram escritos utilizando-se a plataforma **Jupyter Notebook**. Dessa forma, o arquivo onde os algoritmos foram escritos possui a extensão **.ipynb**.

IV. QUESTÕES E SOLUÇÕES

Neste trabalho, foi pedido que aplicássemos filtros passa-alta, passa-baixa e passa-faixa em imagens no espectro de Fourier, para diferentes valores de kernel. Além disso, foi pedido que fizéssemos a compressão de imagens, de acordo com alguma técnica. A seguir, explicamos os métodos empregados.

A. Espectro de Fourier

Em 1822 o matemático francês Jean Baptiste Joseph Fourier mostrou, no livro *The Analytic Theory of Heat*, que qualquer função periódica pode ser expressa como a soma de senos e cossenos de diferentes frequências, os quais são multiplicados por diferentes coeficientes, o que ficou conhecido como **Série de Fourier**. Além disso, também foi mostrado que qualquer função, não necessariamente periódica, que tiver uma área sob a curva finita pode ser expressa como a integral de senos e cossenos que são multiplicados por uma função de ponderação, o que foi chamado de **Transformada de Fourier** [2].

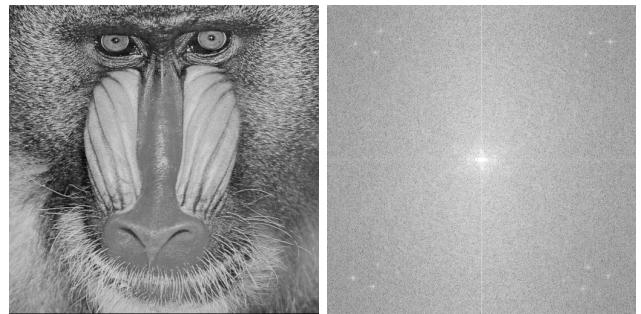
Tanto a série como a transformada de Fourier carregam a importante propriedade de permitir que uma função possa ser reconstruída a partir da série ou transformada dada. Assim, determinadas aplicações são facilitadas ao se trabalhar no domínio de Fourier e converter-se o resultado para o domínio desejado, utilizando as transformações inversas [2].

Para trabalharmos com transformadas e séries de Fourier em imagens digitais, aplicamos o algoritmo conhecido como **Fast Fourier Transform** (FFT), criado nos anos de 1960. Ao aplicarmos esse algoritmo sobre uma imagem digital, ele retorna um *array*, de mesmas dimensões que a imagem, cujos elementos são números complexos. Esses números representam as componentes de frequência da imagem dada e podem ser utilizados para calcular duas características importantes do espectro de Fourier: a magnitude e a fase dos elementos do espectro.

Com a magnitude, conseguimos saber a intensidade de seu respectivo componente de frequência, ou seja, sabemos quais frequências carregam mais energia em um determinado sinal. Já a fase nos diz a direção de cada componente de frequência ou, em outras palavras, no caso de um sinal 1-D, seu deslocamento horizontal. Assim, podemos gerar dois novos *arrays*, com as mesmas dimensões da imagem original, sendo um para os valores de magnitude e o outro para os de fase.

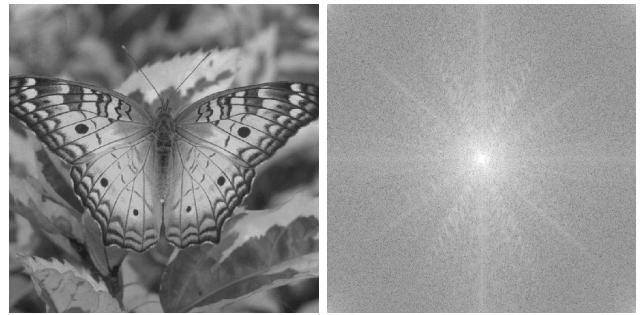
Nas Figuras 1, 2, 3 e 4 podemos observar as imagens originais empregadas neste trabalho e seus respectivos espectros de Fourier, representados pelos *arrays* de magnitude correspondentes.

Para gerar os *arrays* de magnitude, empregamos os métodos *cv.dft()*, da biblioteca *opencv*, e *np.fft.fftshift()* da biblioteca *numpy*. O primeiro método calcula a **Transformada Discreta de Fourier** da imagem de entrada, que é calculada por meio do



(a) Imagem original Baboon. (b) Magnitudes FFT Baboon.

Figura 1: Imagem original Baboon e as magnitudes do seu respectivo espectro de Fourier.



(a) Imagem original Butterfly. (b) Magnitudes FFT Butterfly.

Figura 2: Imagem original Butterffly e as magnitudes do seu respectivo espectro de Fourier.



(a) Imagem original House. (b) Magnitudes FFT House.

Figura 3: Imagem original House e as magnitudes do seu respectivo espectro de Fourier.

algoritmo de FFT. Já o segundo método realiza o deslocamento de componente de frequência zero do espectro para o centro do *array* gerado.

Um exemplo de código para a realização dessas transformações pode ser visto no trecho de código 3.

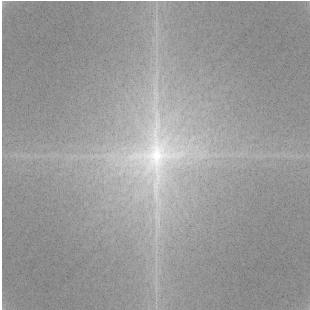
```
dft_image = cv.dft(np.float32(image), flags = cv.DFT_COMPLEX_OUTPUT)
```

```
dft_image = np.fft.fftshift(dft_image)
```

Código 3: Gerar espectro de Fourier e deslocar componente DC para o centro.



(a) Imagem original Seagull.



(b) Magnitudes FFT Seagull.

Figura 4: Imagem original Seagull e as magnitudes do seu respectivo espectro de Fourier.

É interessante notar como os espectros se relacionam com as imagens originais. Na Figura 1b, podemos observar as magnitudes do espectro de Fourier do Baboon. É possível observar uma linha mais clara da direção vertical, a qual pode estar relacionada às inúmeras bordas mais ou menos verticais que encontramos na Figura 1a, principalmente na região do focinho. Além disso, há algumas regiões pequenas de maior intensidade nos cantos superiores e inferiores, direitos e esquerdos, as quais podem ter sido obtidas devido aos pelos e bigodes do Baboon.

Na Figura 2b podemos observar as magnitudes do espectro da Borboleta. Podemos observar que existem linhas de maior intensidade em diversas direções, tanto horizontal e vertical, como em diversos ângulos de diagonal. Essas linhas podem ter sido geradas devido às direções das bordas na Figura 2a. Além disso, algumas das linhas do espectro parecem intercalar regiões mais claras e mais escuras, mostrando que elas podem estar relacionadas à intercalação de intensidades de cinza que acontece, principalmente, nas asas da borboleta.

Já na Figura 3b podemos observar as magnitudes do espectro de Fourier da imagem da Casa. É possível observar que existem linhas mais claras (maior intensidade) nas direções vertical, horizontal e diagonal. Essas linhas podem estar relacionadas às bordas presentes nos tijolos da casa, em suas portas, janelas e escadas, e na árvore, conforme visto na Figura 3a. Além disso, as linhas diagonais do espectro podem estar relacionadas às bordas do telhado da casa.

Por fim, na Figura 4b podemos observar as magnitudes do espectro da Figura 4a. É possível observar que existem duas linhas, uma vertical e outra horizontal, que parecem mais grossas e intercaladas por valores de maior e menor intensidade. Além disso, existem diversas linhas em direções diagonais. Elas podem estar relacionadas principalmente às bordas da pedra sobre as quais os pássaros estão, às janelas no fundo da imagem e às bordas que descrevem o contorno dos pássaros e de suas asas.

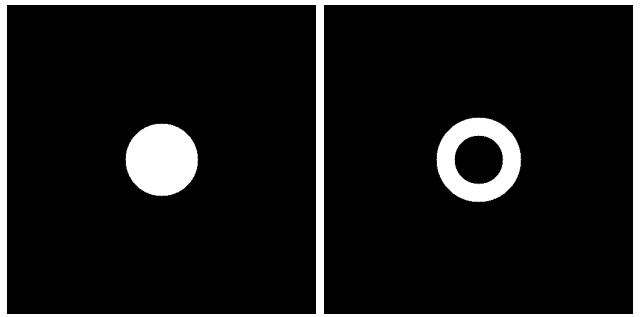
Mais duas observações importantes podem ser feitas sobre as imagens que representam as magnitudes dos espectros de Fourier: a componente de frequência zero, que foi transladada para o centro da imagem, representa a média das intensida-

des presentes na imagem original; além disso, quanto mais próximos do centro, menores são as frequências que os valores representam e, consequentemente, quanto mais distantes do centro do array maiores são as frequências representadas [2]. Esta última observação vai nos ajudar no entendimento dos filtros criados, os quais serão explicados nas próximas seções.

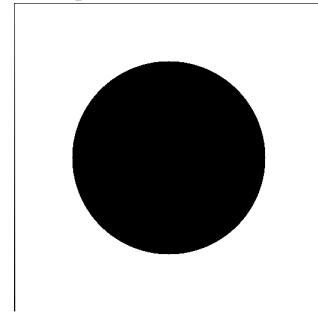
B. Construção e aplicação dos filtros

Na implementação deste trabalho, cada filtro foi construído como um array *numpy*. Além disso, todos os filtros criados são ideais, ou seja, há uma transição abrupta entre a região de corte e a região que deve ser mantida intacta.

Dada a configuração do espectro de Fourier explicada na seção anterior, em que as frequências mais baixas ficam próximas do centro da imagem e as frequências mais altas ficam mais distantes, os filtros gerados são *arrays* que formam um círculo de determinado raio cujos valores de *pixels* (zero ou um) dependem do tipo de filtro desejado. Nas Figuras 5a, 5b e 5c podemos observar um exemplo de filtro passa-baixa (LPF), passa-banda (BPF) e passa-alta (HPF), respectivamente. É importante destacar que todos os filtros têm as dimensões da imagem original e que as bordas pretas ao redor do filtro passa-alta são apenas ilustrativas, ou seja, foram colocadas para que os limites da imagem pudessem ser vistos no relatório.



(a) LPF de raio 60 pixels. (b) BPF de raios 70 e 40 pixels.



(c) HPF de raio 160 pixels.

Figura 5: Exemplos de filtros ideais utilizados no trabalho.

Neste trabalho, foram gerados vinte filtros passa-baixa e vinte passa-alta, de raios $\{10, 30, 60, 160, 200\}$, sendo cinco filtros para cada imagem. Já os filtros passa-banda foram gerados pelas combinações de raios pequenos de dimensões $\{10, 20, 30, 40, 50\}$ e raios grandes de dimensões $\{60, 70, 80, 90\}$, sendo vinte filtros para cada imagem e oitenta no total.

Conforme visto nas aulas, um filtro pode ser aplicado a uma imagem por meio do processo de convolução. No domínio de frequência, a convolução entre dois sinais pode ser realizada pela sua multiplicação. Assim, para realizarmos a filtragem das imagens utilizadas neste trabalho, primeiro as colocamos no domínio de frequência, através da transformada de Fourier, e então aplicamos os filtros, multiplicando-os pelos espectros de Fourier gerados.

No trecho de código 4 podemos observar um exemplo da aplicação de um filtro a uma imagem, por meio de sua multiplicação pelo espectro de Fourier gerado a partir da imagem original.

```
filtered_image = dft_image * ideal_filter
```

Código 4: Exemplo de convolução no domínio da frequência.

Após a filtragem da imagem, transladamos a componente de frequência zero do espectro resultante do centro para o canto superior esquerdo e então aplicamos a transformada inversa de Fourier, obtendo a imagem resultante no domínio espacial.

Os filtros gerados neste trabalho são armazenados em um dicionário chamado *ideal_kernels*, cujas chaves indicam o tipo de filtro e a imagem sobre a qual ele será aplicado. Além disso, cada chave possui como valor uma lista, em que cada elemento é um dicionário cujo valor é o filtro gerado e a chave indica seu raio. Assim, um filtro passa-baixa, de raio 10, a ser aplicado na imagem do *Baboon* estará na lista de chave "LPF_Baboon" no elemento de chave "10".

Cada imagem é armazenada em um dicionário chamado *images*, cujas chaves são os nomes das imagens. Por exemplo, a imagem da borboleta pertencerá à chave "Butterfly". Também, os espectros de Fourier gerados a partir das imagens de entrada são armazenados em um dicionário chamado *dft_images*, em que as chaves são os nomes das imagens originais.

Os espectros resultantes da filtragem são armazenados em um dicionário chamado *image_spectrums*, que possui a mesma estrutura do dicionário *ideal_kernel*, ou seja, cada um de seus elementos possui uma chave que indica o tipo de filtro e a imagem, e eles são listas de dicionários que indicam o raio dos filtros.

Por fim, as imagens convertidas para o domínio espacial são armazenadas em um dicionário chamado *filtered_images*, em que a chave indica o tipo de filtro, a imagem em que ele foi aplicado e o raio do filtro aplicado.

C. Compressão das imagens

A compressão é uma importante ferramenta de processamento de imagens, que permite que representemos uma imagem com uma menor quantidade de dados, mas mantendo a informação que ela carrega.

Neste trabalho, aplicamos dois processos de compressão através da remoção de componentes (coeficientes), atribuindo valor zero a elas, de acordo com dois critérios: o primeiro deles foi um valor de limiar, de maneira que toda componente menor do que esse valor recebeu valor zero; o segundo critério foi uma porcentagem da quantidade de componentes.

Neste último, ordenamos o *array* de componentes em ordem crescente de magnitudes e zeramos, nessa ordem, uma determinada quantidade, que foi obtida como uma porcentagem da quantidade total de componentes.

Ambas as operações foram aplicadas sobre o espectro de Fourier das imagens. Assim, após a compressão, fizemos a translação da componente DC para o canto superior esquerdo e aplicamos a transformada inversa de Fourier de maneira a levar a imagem para o domínio espacial.

Para a compressão baseada em um valor limitante, os valores usados como limite foram {120, 140, 160, 180, 200, 220, 240, 260}. Já para aquela baseada em uma porcentagem da quantidade de coeficientes, as porcentagens escolhidas, em valores decimais, foram {0.5, 0.8, 0.9, 0.95, 0.99, 0.995, 0.999, 0.9995}.

V. RESULTADOS E DISCUSSÕES

Nos próximos parágrafos, apresentamos os testes realizados e os resultados obtidos para cada um dos tipos de filtro e para os métodos de compressão empregados.

A. Filtros passa-baixa

Filtros passa-baixa, ou de suavização, são empregados para a diminuição da presença de transições de intensidade abruptas em imagens. Isso permite que eles sejam utilizados para o borrimento de imagens, de maneira a reduzir detalhes irrelevantes e facilitar a extração de objetos, para a redução de ruídos e para o preenchimento de lacunas.

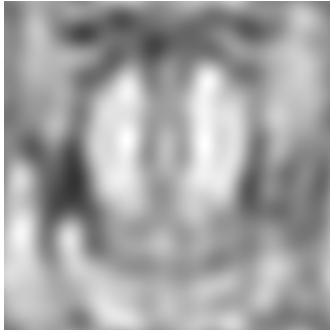
Entretanto, como um efeito colateral indesejado está o borrimento das bordas, que são características importantes de uma imagem e também são representadas por transições nítidas de intensidade. Além disso, quando aplicamos filtros ideais no domínio de frequência surge um efeito conhecido como *Ringing* ou *Efeito de Gibbs*, que faz com que apareçam linhas ao longo da imagem que repetem o padrão de bordas próximas [2].

Na Figura 6 podemos observar o resultado da aplicação de filtros passa-baixa na Figura 1a, para os diferentes valores de raio. Já na Figura 7 podemos observar os resultados dos filtros passa-baixa aplicados na Figura 2a.

Conforme pode-se observar nas Figuras 6 e 7, quanto maior é o raio do filtro passa-baixa empregado, menos atenuada se torna a imagem, pois mais frequências são mantidas no espectro. Isso acontece porque o círculo formado pelo filtro, como o da Figura 5a, mantém as frequências que estão dentro dele.

Também, conforme visto nas Figuras 6a e 7a, um valor pequeno de raio gera uma grande atenuação na imagem original, de maneira que os objetos presentes nela podem se tornar indistinguíveis.

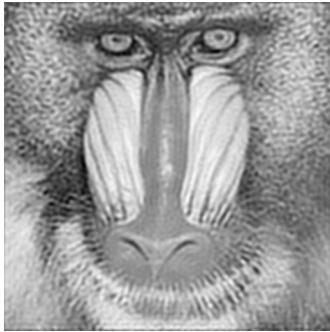
Além disso, conforme esperado, podemos observar linhas que não existiam nas imagens originais e que apareceram devido ao efeito de Gibbs, como pode ser visto nas Figuras 7c e 7d. Também pode-se perceber que, conforme aumenta-se o raio do filtro, menor se torna o efeito de Gibbs produzido.



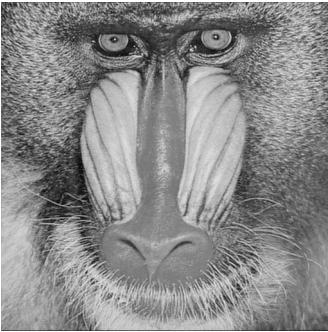
(a) LPF Baboon raio 10.



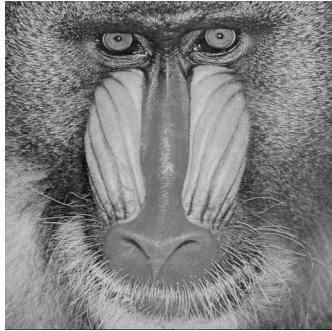
(b) LPF Baboon raio 30.



(c) LPF Baboon raio 60.



(d) LPF Baboon raio 160.



(e) LPF Baboon raio 200.

Figura 6: Resultados dos LPF no Baboon para diferentes raios.

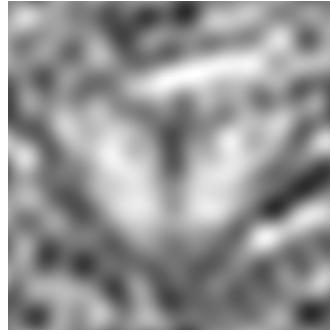
B. Filtros passa-alta

Os filtros passa-alta, ou de nitidez, são utilizados para destacar regiões de transição de intensidade, enfatizando bordas e outras descontinuidades, e eliminando áreas em que as intensidades mudam lentamente [2].

Os filtros desse tipo costumam gerar descontinuidades acinzentadas sobre fundos pretos (sem informação). Uma maneira de recuperar a informação contida no fundo, é somar a imagem resultante da filtragem à imagem original [2].

Apesar disso, esse tipo de filtro pode gerar o efeito negativo de destacar ruídos. Ademais, filtros passa-alta ideais, no domínio de frequência, também sofrem com o efeito de Gibbs, fazendo com que os objetos da imagem tenham bordas distorcidas e grossas. Todavia, quanto maior for o raio do filtro empregado, menor se torna o efeito de Gibbs produzido.

Nas Figuras 8 e 9 podemos observar os resultados da aplicação de filtros passa-alta, com diferentes raios, sobre as



(a) LPF Butterfly raio 10.



(b) LPF Butterfly raio 30.



(c) LPF Butterfly raio 60.



(d) LPF Butterfly raio 160.



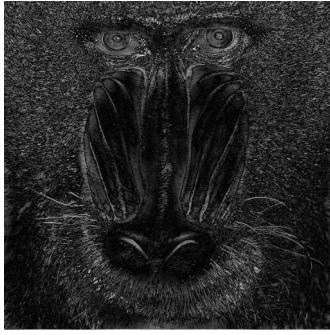
(e) LPF Butterfly raio 200.

Figura 7: Resultados dos LPF na Butterfly para diferentes raios.

Figuras 1a e 2a.

Conforme podemos observar nas figuras, a filtragem de frequências altas destacou as bordas e outras descontinuidades das imagens. Ao contrário do que aconteceu com alguns dos filtros passa-alta utilizados no Trabalho 02, como os operadores de sobel e os filtros de diagonal, as imagens resultantes dos filtros aplicados neste trabalho tiveram descontinuidades destacadas em todas as direções.

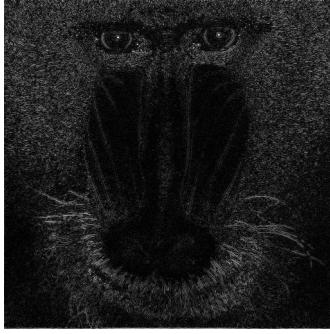
Além disso, podemos perceber o efeito de Gibbs descrito, como pode ser observado nas Figuras 8a e 9b, em que as imagens possuem linhas que não existiam nas imagens originais e certas bordas se tornaram mais grossas e distorcidas, gerando regiões que parecem ter sido preenchidas por pixels de maior intensidade, como as bolinhas das asas da borboleta e o nariz do Baboon. Conforme aumentamos o raio do filtro passa-alta, menos frequências são mantidas, e menor é o efeito de Gibbs percebido.



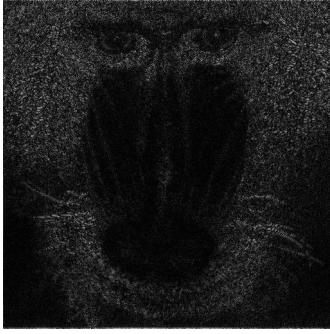
(a) HPF Baboon raio 10.



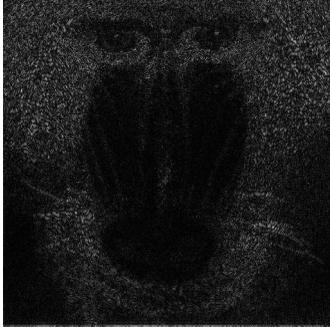
(b) HPF Baboon raio 30.



(c) HPF Baboon raio 60.



(d) HPF Baboon raio 160.



(e) HPF Baboon raio 200.

Figura 8: Resultados dos HPF no Baboon para diferentes raios.

Apesar disso, assim como os detalhes das imagens (como bordas), os ruídos também estão presentes nas frequências mais altas. Assim, quando o raio do filtro é muito grande, as imagens resultantes podem ter mais ruído do que informações úteis. Nas Figuras 8e e 9e podemos observar exemplos em que uma situação semelhante acontece, embora ainda seja possível distinguir as formas e alguns detalhes dos animais representados.

C. Filtros passa-banda

Filtros passa-banda (ou passa-faixa), por sua vez, são empregados para processar determinada banda do espectro de frequências. Isso os permite serem empregados para diversos tipos de aplicações, como por exemplo, a remoção de determinados efeitos indesejados em uma imagem, decorrentes das características de determinadas faixas de frequências presentes nessa imagem.

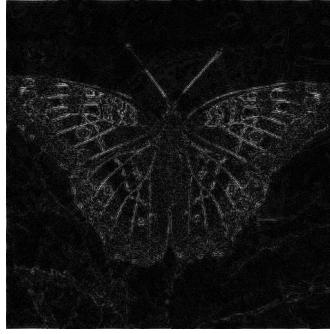
Nas Figuras 10 e 11 podemos observar os resultados da



(a) HPF Butterfly raio 10.



(b) HPF Butterfly raio 30.



(c) HPF Butterfly raio 60.



(d) HPF Butterfly raio 160.



(e) HPF Butterfly raio 200.

Figura 9: Resultados dos HPF na Butterfly para diferentes raios.

aplicação de filtros passa-banda de raios $\{(60, 10), (60, 20), (60, 30), (60, 40), (60, 50)\}$ e $\{(80, 10), (80, 20), (80, 30), (80, 40), (80, 50)\}$, respectivamente, nas Figuras 1a e 2a.

Podemos observar nas figuras que a preservação de apenas uma determinada faixa de frequências, para os valores de raio adotados, faz com que os filtros passa-banda gerem imagem semelhantes àquelas resultantes de filtros passa-alta. Ou seja, os filtros passa-banda, nos casos mostrados, também permitiram destacar os detalhes das imagens.

Apesar disso, como empregamos filtros passa-banda ideais, eles também sofreram mais fortemente com o efeito de Gibbs. Podemos perceber que quanto mais o raio menor se aproxima do maior, ou seja, quanto menor é a faixa de frequências mantida intacta, mais distorcidas se tornam as imagens resultantes, tornando-se difícil distinguir as objetos que elas possuem, como acontece na Figura 10e.

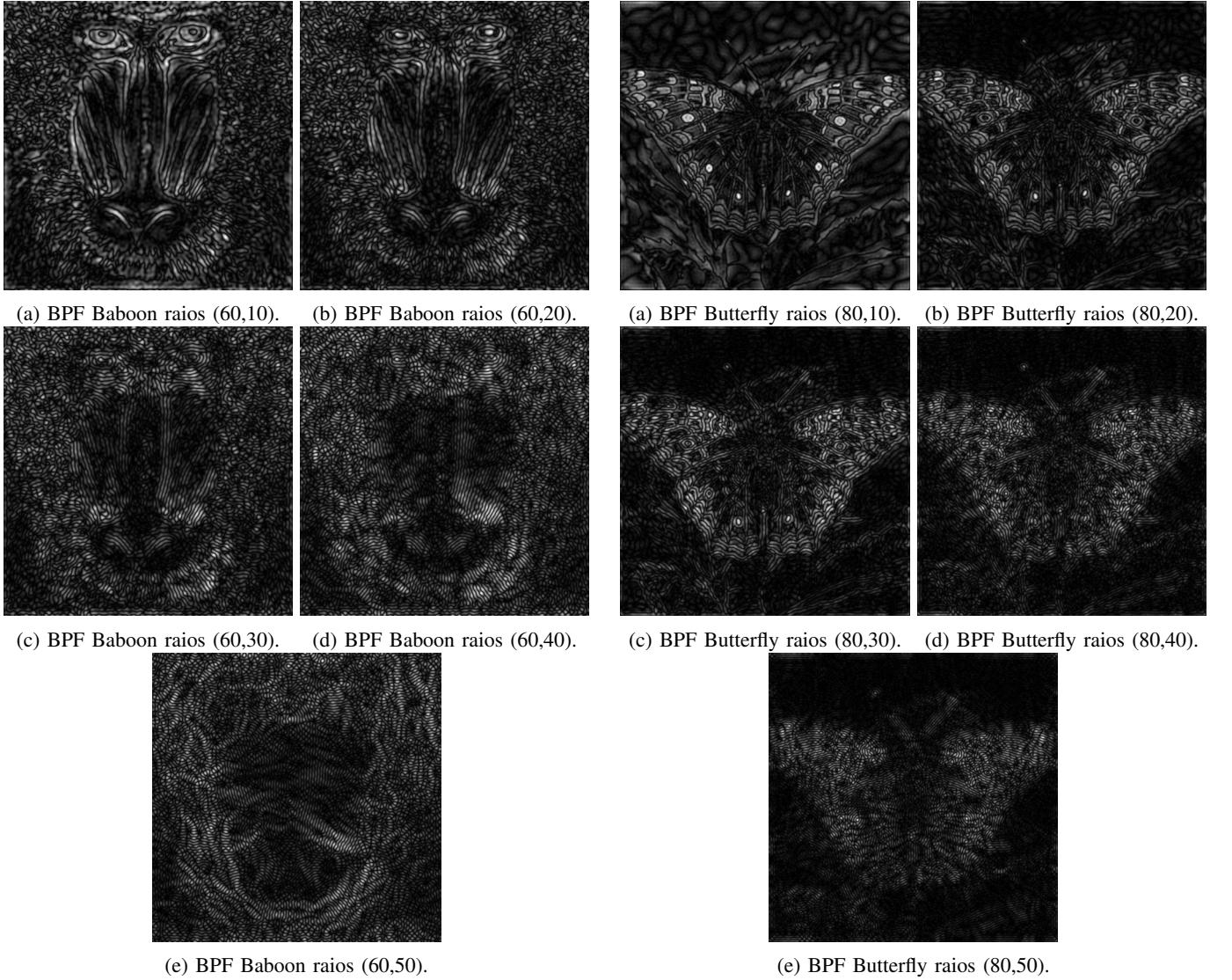


Figura 10: Resultados dos BPF no Baboon para diferentes raios.

D. Compressão de imagens

Conforme explicado na Seção IV-C, a compressão das imagens foi realizada de duas maneiras diferentes. É importante destacar que as magnitudes calculadas para os coeficientes do espectro de Fourier das imagens de entrada são apresentadas na escala logarítmica. Na Figura 12 podemos observar os resultados da compressão ao empregarmos a técnica por valor limitante.

Podemos observar que, até o limitante 180, não conseguimos perceber que a imagem original foi alterada. Isso pode ser devido ao fato de o número de coeficientes zerados ser pequeno em relação à quantidade total, ou seja, a quantidade de coeficientes cujos valores são menores do que 180 é relativamente pequena se comparada à quantidade total de coeficientes.

Por outro lado, a partir do limitante 200, conseguimos per-

ceber que a imagem gerada difere-se da imagem original e que sua qualidade diminui, tornando-se cada vez mais distorcida. Em outras palavras, conforme aumentamos o valor do limitante, mais coeficientes zeramos e menos informação sobre a imagem original é mantida. Ao empregarmos o limitante 240, conseguimos obter uma imagem próxima a que foi dada como exemplo na definição do trabalho. Ao empregarmos o limitante 260, o animal presente na imagem torna-se irreconhecível, mostrando que grande maioria dos coeficientes possui uma magnitude menor do que 260.

Na Figura 13, podemos observar os resultados da compressão ao empregarmos a porcentagem da quantidade total de coeficientes.

Como estamos zerando uma determinada quantidade de coeficientes em ordem crescente, zeramos sempre os coeficientes que têm menor magnitude. Podemos notar que, conforme aumentamos a porcentagem de coeficientes a serem zerados,

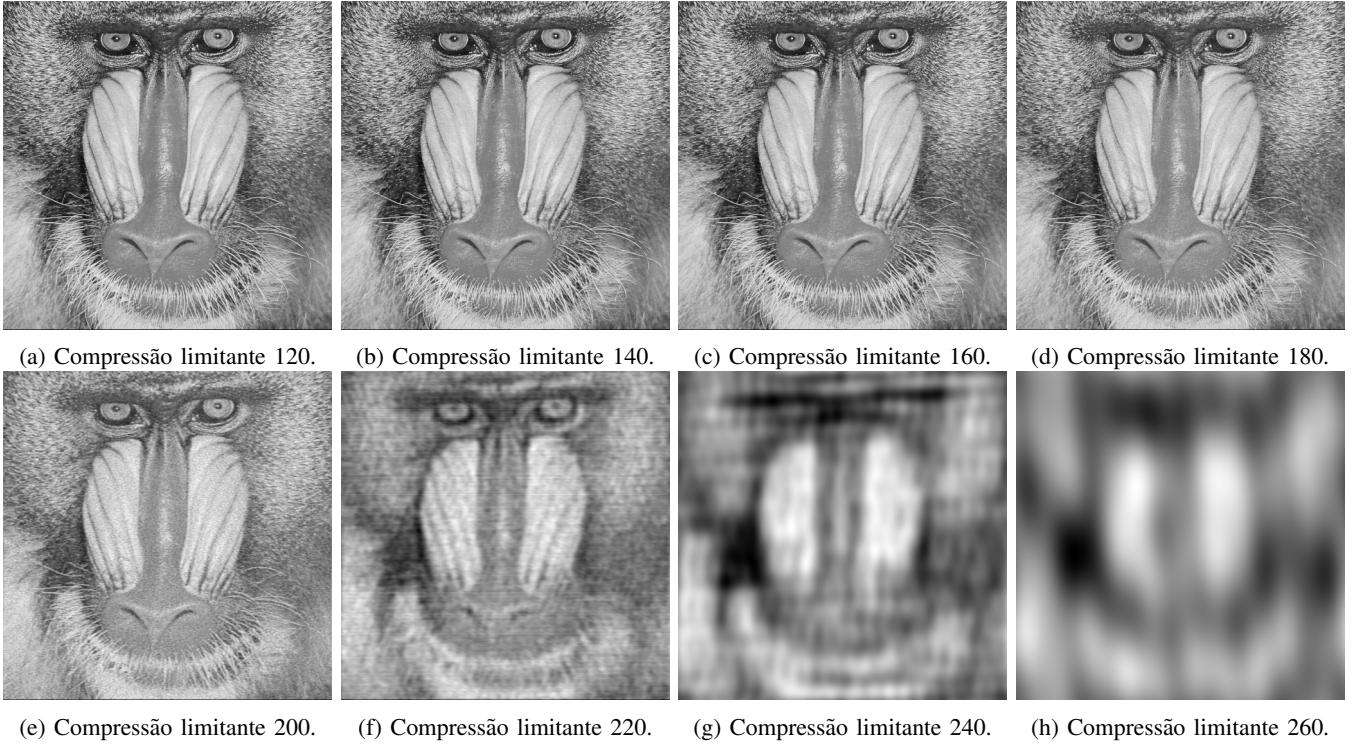


Figura 12: Resultados de compressão no Baboon usando valor limitante.

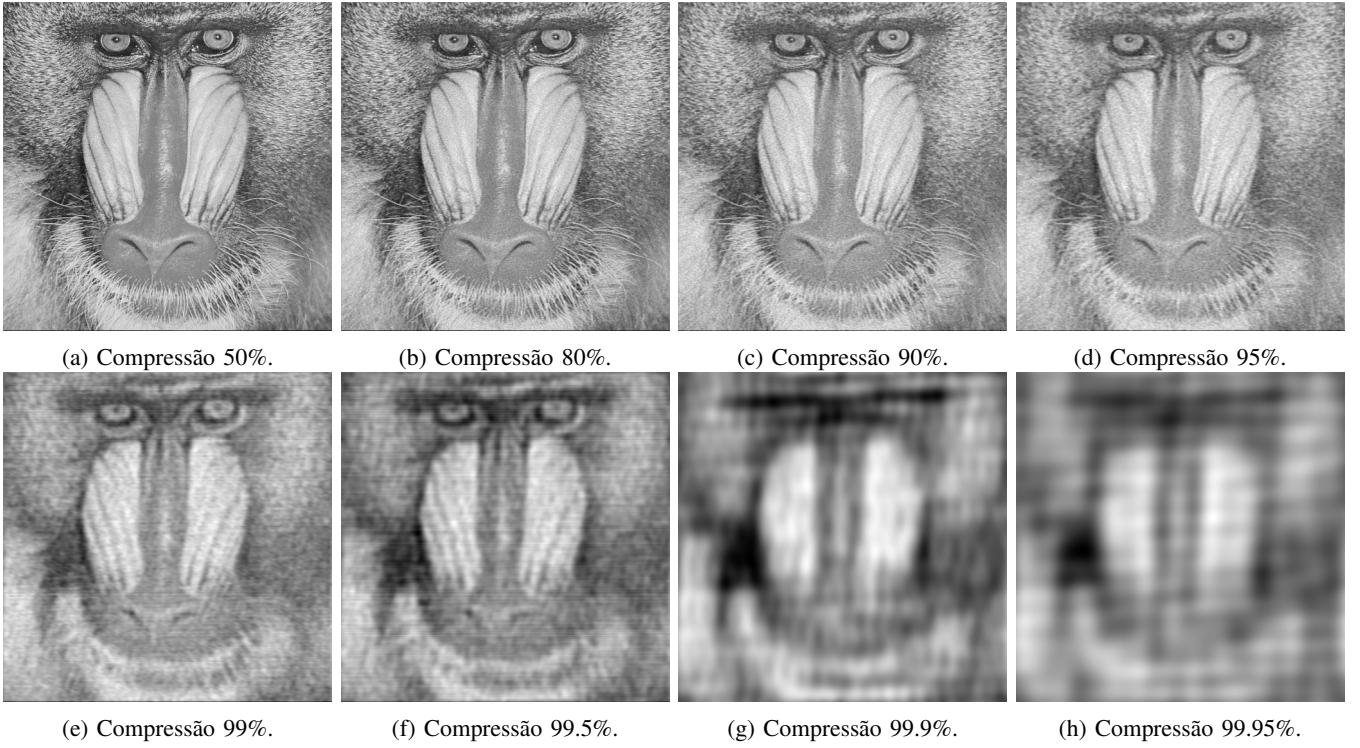


Figura 13: Resultados de compressão no Baboon usando porcentagem de coeficientes.

mais distorcida se torna a imagem.

Ao observarmos a Figura 13a, percebemos que a imagem original aparenta não ter sido alterada, ou seja, a imagem

resultante é semelhante à original. Assim, comparando com os resultados da Figura 12, podemos perceber que aproximadamente 50% dos coeficientes possui magnitude menor que

Também, ao compararmos as Figuras 13c e 12e, percebemos que, aproximadamente, 90% dos coeficientes possuem magnitude menor que 200. Além disso, observamos na Figura 13g que obtivemos uma imagem semelhante à apresentada na descrição do trabalho quando zeramos 99.9 % dos coeficientes do espectro de Fourier da figura utilizada como entrada.

VI. CONCLUSÃO

Conforme pôde ser observado, conseguimos fazer o que foi pedido no Trabalho 03, limitando-nos, entretanto, a imagens monocromáticas. Foram aplicados cinco filtros passa-alta, cinco filtros passa-baixa e vinte filtros passa-banda sobre quatro imagens diferentes.

Para gerarmos novos filtros de um determinado tipo, apenas alteramos o valor de seu raio. Além disso, todos os filtros produzidos são considerados ideais. Apesar de esses filtros não serem empregados na prática, seu estudo nos permite entender conceitos importantes de filtragem no domínio da frequência.

Para realizar a filtragem no domínio da frequência, aplicamos a convolução, que nesse domínio pode ser feita por meio de uma multiplicação. Assim, os seguintes passos foram realizados: aplicamos a transformada de Fourier sobre as imagens originais; fizemos a translação das componentes DC para o centro de cada imagem; realizamos a multiplicação entre os espectros de Fourier e os filtros (convolução); fizemos a translação da componente de frequência zero dos espectros resultantes para o canto superior esquerdo; e aplicamos a transformada inversa de Fourier para levar as imagens de volta para o domínio espacial.

Conforme podemos observar, os filtros passa-baixa atenuaram as imagens, os filtros passa-alta destacaram as descontinuidades e os filtros passa-banda, nas faixas escolhidas, também destacaram descontinuidades. Apesar disso, os três tipos de filtro sofreram com um efeito conhecido como *ringing* ou *Efeito de Gibbs*, de maneira que para determinados valores de raio, as imagens resultantes tornaram-se bastante distorcidas e apresentaram detalhes que não existiam nas originais. Ademais, as imagens resultantes dos filtros passa-banda aparentaram ser as que mais sofreram com esse efeito.

Com relação à compressão de imagens, conseguimos atingir níveis de compressão desejados nos dois métodos empregados, podendo inclusive estabelecer uma relação entre os valores dos coeficientes e a porcentagem da quantidade dos coeficientes que possuíam esses valores.

REFERÊNCIAS

- [1] H. Pedrini and W. R. Schwartz, *Análise de imagens digitais: princípios, algoritmos e aplicações*. Thomson Learning, 2008. 1
- [2] R. C. Gonzalez, R. E. Woods, and B. R. Masters, “Digital image processing third edition,” *Pearson International Edition*, 2008. 1, 2, 3, 4, 5