

# Relatório do Trabalho 1 de Introdução ao Processamento de Imagens

Pedro Nogueira

*Departamento de Ciência da Computação*

*Universidade de Brasília*

Brasília, Brasil

cmcg89034@gmail.com

**Resumo**—A matéria Introdução a Processamento de Imagens, ministrada na Univesidade de Brasília, pelo professor Bruno Macchiavello, tem por objetivo ensinar a seus alunos diversas operações que são possíveis fazer em imagens. O objetivo da matéria é de formar alunos capazes de aplicar essas operações para diversos fins, desde apagar ruídos de imagens até reconhecimentos sofisticados em suas características visuais. O objetivo desse primeiro projeto é avaliar o entendimento dos estudantes nesse primeiro momento de avaliações da matéria.

**Index Terms**—MATLAB, Imagens YUV/RGB, Suavização de píxels por vizinhança, Filtros de convolução, Filtros laplaciano e gaussiano, Imagens no domínio da frequência, Filtros de passa-altas/passa-baixas no domínio da frequência, Filtros rejeita-notch.

## I. INTRODUÇÃO

A disciplina começa apresentando processamentos de baixo nível em imagens, que são processamentos que recebem uma imagem como entrada e retornam a própria imagem com algumas modificações. Este primeiro trabalho cobre parte dos conteúdos passados nesse primeiro momento, que são:

- Alargamento e suavização de imagens por vizinhança de píxels;
- Aplicação de filtros laplacianos e gaussianos;
- Desenvolvimento de filtros rejeita-notch.

A primeira parte do projeto traz os conceitos de vídeo quadro por quadro, formato de cores YUV, e alargamento e suavização. Um vídeo disposto quadro por quadro é simplesmente uma sequência de bytes que contém cada um dos quadros do vídeo, sendo cada um desses quadros uma imagem correspondente àquele momento do vídeo. O formato de cores YUV, diferente do RGB (**R**ed **G**reen **B**lue), tem por sua composição três imagens, uma contendo o brilho e as outras duas a composição geral das cores da imagem original.

A segunda parte do projeto contém os conceitos de filtros e convoluções no domínio do espaço. Ao se tratar uma imagem como um sinal no domínio do espaço, de coordenadas bidimensionais horizontal e vertical, pode ser feita uma convolução na imagem usando o núcleo desejado, bastando inverter esse núcleo e passá-lo por cada píxel da imagem pra se fazer a média de suas intercessões em cada posição. Os filtros laplaciano e gaussiano são simplesmente filtros que foram elaborados em sua teoria pelos matemáticos Pierre-Simon Laplace e Carl Friedrich Gauss respectivamente.

A terceira parte do trabalho usa os conceitos de imagens no domínio da frequência, filtro passa-altas, filtro Butterworth, e filtros rejeita-notch. Tratando uma imagem como um sinal e aplicando a transformada de Fourier discreta e não-periódica nela, vinda da série calculada pelo matemático Jean-Baptiste Joseph Fourier, obtém-se assim a imagem no domínio da frequência. Essa imagem contém informações das frequências da imagem original, sendo as baixas frequências representativas do formato geral da imagem e as altas frequências representativas das bordas da imagem. Um filtro passa-altas nada mais significa que a aplicação de uma máscara binária ou gradiente que anula baixas frequências da imagem. O filtro Butterworth utilizado no projeto é um filtro calculado pelo engenheiro Stephen Butterworth, e um filtro rejeita-notch é o conjunto de vários filtros passa-altas com seus centros em posições desejadas. Um filtro rejeita-notch é muito útil para a eliminação do efeito de padrões Moiré de uma imagem, o que será executado nesse projeto.

## II. METODOLOGIA

### A. Imagens YUV e alargamento de píxels

O arquivo `foreman.yuv` foi disponibilizado para o teste da primeira parte do projeto, que é composto por vários bytes que representam cada um dos quadros de um vídeo. Esse vídeo tem por resolução 352x288, e suas componentes YUV são disponibilizadas no formato 4:2:0, o que significa que a componente Y está na resolução 352x288, enquanto as componentes U e V estão cada uma na metade da resolução em ambos os eixos, ficando então em resolução 176x144. Considerando  $b$  uma quantidade de bytes,  $w$  a largura de uma imagem, e  $h$  a altura de uma imagem, um quadro  $Q$  então contém a quantidade de bytes calculada na equação 1.

$$Q_b = Y_h \times Y_w + 2 \times (U_h \times U_w) + 2 \times (V_h \times V_w) \quad (1)$$

Para se obter um quadro  $N$  (começando em zero) no arquivo, foi preciso primeiro calcular quantos bytes seriam ignorados para então começar a leitura, e isso é equivalente à quantidade de bytes de  $N$  quadros.

$$Q_{inicio} = N \times Q_b \quad (2)$$

Tabela I  
PÍXELS ORIGINAIS

|   |    |    |
|---|----|----|
| 2 | 4  | 6  |
| 8 | 10 | 12 |

Tabela II  
PÍXELS REPLICADOS

|   |   |    |    |    |    |
|---|---|----|----|----|----|
| 2 | 2 | 4  | 4  | 6  | 6  |
| 2 | 2 | 4  | 4  | 6  | 6  |
| 8 | 8 | 10 | 10 | 12 | 12 |
| 8 | 8 | 10 | 10 | 12 | 12 |

Como as componentes da imagem estão em dimensões diferentes, foi preciso dobrar a altura e largura das componentes U e V. O professor propôs dois métodos de alargamento: o de replicar cada píxel em um quadrado de 4 píxels 2x2, e o de preencher os píxels novos criados com as médias dos píxels vizinhos, se disponíveis. Um exemplo de um conjunto de seis píxels 3x2 se encontra na tabela I, e seus dois tipos de alargamento diferentes nas tabelas II e III.

Dobradas as dimensões dos componentes U e V, pode-se gerar a imagem RGB ao se usar a função `ycbcr2rgb()` do MATLAB.

Para dobrar o tamanho total da imagem, foram aplicados os algoritmos de alargamento no componente Y original e nos componentes U e V previamente alargados. Feito isso, as componentes novamente estão em dimensões iguais, sendo possível então usar a função `ycbcr2rgb()` novamente.

#### B. Filtragem de convolução no domínio do espaço

A segunda parte do projeto prevê que o aluno deve fazer filtrações por convolução na imagem `Imagem1.pgm` no domínio do espaço, ou seja, na própria imagem. A convolução das imagens foram todas feitas usando núcleos específicos com a função `imfilter()`. Foram usados quatro núcleos diferentes para essas filtrações: dois foram obtidos com a função do MATLAB `fspecial()` com os parâmetros 'gaussian' para o modo do núcleo e 3 para a dimensão do núcleo, mudando somente o terceiro parâmetro  $\sigma$  para 0.5 ou para 1 quando requisitado; e dois desses núcleos são os laplacianos de tamanho 3x3 e centro -4 e -8 representados nas tabelas IV e V respectivamente.

Tabela III  
PÍXELS EM MÉDIA

|   |   |    |    |    |    |
|---|---|----|----|----|----|
| 2 | 3 | 4  | 5  | 6  | 6  |
| 5 | 6 | 7  | 8  | 9  | 6  |
| 8 | 9 | 10 | 11 | 12 | 12 |
| 8 | 8 | 10 | 10 | 12 | 12 |

Tabela IV  
LAPLACIANO CENTRO -4

|   |    |   |
|---|----|---|
| 0 | 1  | 0 |
| 1 | -4 | 1 |
| 0 | 1  | 0 |

Tabela V  
LAPLACIANO CENTRO -8

|   |    |   |
|---|----|---|
| 1 | 1  | 1 |
| 1 | -8 | 1 |
| 1 | 1  | 1 |



Figura 1.

Resultado da aplicação do filtro laplaciano em `Imagem1.pgm`

A aplicação do filtro gaussiano suaviza a imagem. O filtro laplaciano cria uma imagem com as bordas e contornos da original, que é usada para subtrair da imagem original e obter uma imagem com os contornos destacados. A figura 1 é exemplo de uma imagem obtida pela aplicação direta do filtro laplaciano 3x3 com centro -8. Então, três procedimentos foram executados:

Primeiro foi pedido que se fizesse a convolução com o filtro laplaciano de centro -8 diretamente.

Depois foi pedido que fosse aplicado primeiramente um filtro gaussiano de  $\sigma = 0.5$  para depois aplicar um filtro laplaciano de centro -4.

Por último, foi aplicado um filtro gaussiano de  $\sigma = 1$  para depois aplicar um filtro laplaciano de centro -4.

#### C. Filtragem rejeita-notch no domínio da frequência

Na terceira e última parte do projeto foi requisitado que fosse feita uma filtragem rejeita-notch de frequências especificadas pelas instruções do projeto na imagem `moire.tif`, que tem padrões de Moiré extremamente evidentes pela imagem toda.

Para isso, foi primeiramente necessário transformar a imagem para o domínio da frequência usando a função do MATLAB `fft2()` e reposicionar os quadrantes da imagem de forma que as baixas frequências fiquem no centro da imagem e as altas nos cantos e bordas usando a função `fftshift()`.

O filtro rejeita-notch requisitado pelas especificações foi o filtro Butterworth, filtro passa-altas que gradualmente vai de zero a um conforme a distância  $D(k,l)$  do centro do filtro até um raio  $D_0$  com um expoente  $n$ . A equação do gradiente Butterworth é dada na equação 3.

$$H(k,l) = \sqrt{\frac{1}{1 + (\frac{D_0}{D(k,l)})^{2n}}} \quad (3)$$

O projeto especificou as posições relativas ao centro da imagem transformada e os raios dos filtros de Butterworth a serem aplicados nessas posições, e esses dados são:

- $D_0 = 10$ ,  $u_k = 39$ ,  $v_k = 30$
- $D_0 = 10$ ,  $u_k = -39$ ,  $v_k = 30$

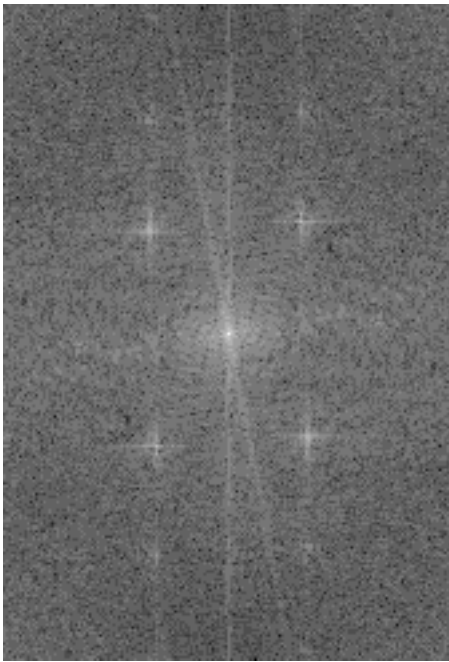


Figura 2.

Imagem `moire.tif` no domínio da frequência com os quadrantes reposicionados

- $D_0 = 5$ ,  $u_k = 78$ ,  $v_k = 30$
- $D_0 = 5$ ,  $u_k = -78$ ,  $v_k = 30$

Essas posições  $u_k$  e  $v_k$  relativas ao centro são os locais onde se deve aplicar os filtros rejeita-notch. Entretanto, por definição, uma imagem no domínio da frequência carrega a mesma frequência em dois pontos, onde ambos são opostos verticalmente e horizontalmente, o que significa que esse filtro Butterworth precisa ser aplicado em oito posições. Foi criada então uma máscara contendo esses filtros de Butterworth, representada pela figura 3, para que essa fosse aplicada de uma só vez na imagem no domínio da frequência.

A aplicação de filtros deve ser feita por convolução no domínio do espaço, mas há uma facilidade no domínio da frequência para essa aplicação, pois ela se resume à simples multiplicação da máscara e da transformada. Feita a filtragem, basta fazer o reposicionamento dos quadrantes da imagem de volta pra as posições iniciais com `fftshift()` e aplicar a transformada de Fourier novamente com `fft2()`.

### III. RESULTADOS

#### A. Imagens YUV e alargamento de pixels

Duas imagens finais foram geradas, uma com alargamento por replicação de pixels (figura 4) e outra por média de pixels vizinhos (figura 5).

Após tanto alargar os componentes U e V responsáveis pelas cores da imagem, esperava-se que algum defeito ficasse mais aparente, mas o que mais interferiu na imagem foi a modificação na componente de brilho Y. O método de alargamento de pixel por cálculo de média dos vizinhos é

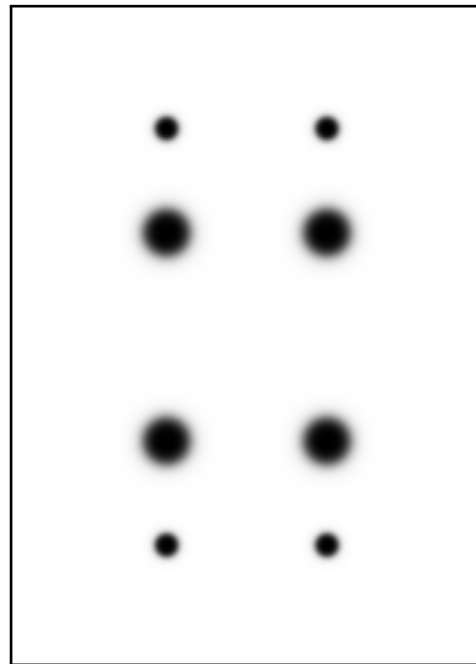


Figura 3.

Máscara contendo os filtros de Butterworth



Figura 4.

Imagem YUV alargada por replicação



Figura 5.

Imagem YUV alargada por cálculo de média



Figura 6.

Imagem com aplicação do filtro laplaciano 3x3 de centro -8



Figura 7.

Imagem com aplicação do filtro gaussiano de  $\sigma 0.5$  e laplaciano 3x3 de centro -4

subjetivamente melhor que o método de simples replicação dos pixels, o que é aparente principalmente no texto no canto superior da esquerda "SIEMENS".

#### B. Filtros de convolução no domínio do espaço

Os três procedimentos diferentes geraram as três imagens representadas pelas figuras 6, 7 e 8.

O primeiro procedimento, por ter sido somente uma filtragem passa-altas, foi muito abrupto em sua execução, deixando



Figura 8.

Imagem com aplicação do filtro gaussiano de  $\sigma 1$  e laplaciano 3x3 de centro -4



Figura 9.

Imagem moire.tif após aplicação do filtro rejeita-notch

a imagem com as bordas muito acentuadas. Os próximos procedimentos fizeram uma filtragem passa-altas também, mas só depois de fazer uma suavização da imagem, tornando-a mais coerente para visualização.

#### C. Filtragem rejeita-notch no domínio da frequência

Inicialmente, moire.tif, a imagem usada nessa parte do projeto, continha padrões de Moiré muito proeminentes na forma de diversos "quadrados" pela imagem, e como indicado no resultado do procedimento (figura 9), esse padrão desapareceu da maioria da imagem. Mais próximo das bordas ainda há o efeito Moiré, e isso se dá por não ter sido feito padding na imagem no domínio da frequência antes de fazer sua filtragem rejeita-notch.

### IV. CONCLUSÕES

Todos os procedimentos desse primeiro trabalho foram realizados com sucesso. Os resultados dos procedimentos foram esperados por terem sido assunto das aulas de Introdução ao Processamento de Imagens, mas ainda assim foi surpreendente testemunhar a gama de possibilidades em que uma imagem pode ser processada. O projeto foi um sucesso em sua execução como um todo, portanto as aulas também foram um sucesso no ensinamento de seus docentes.

### REFERÊNCIAS

- [1] Slides das aulas de Introdução ao Processamento de Imagens do professor Bruno Macchiavello, Universidade de Brasília.