

Introdução ao Processamento de Imagens

Projeto 1

Cássio Borges
Departamento de Ciência da Computação
Universidade de Brasília
Brasília, Brasil
211036141@aluno.unb.br

Abstract—A disciplina de Introdução ao Processamento de Imagens tem como objetivo ensinar técnicas para aprimorar a visualização de imagens aos olhos humanos e utilizar computadores para uma melhor compreensão do mundo ao nosso redor. Durante o curso, os alunos adquirem um entendimento aprofundado dos componentes que formam as imagens, aprendem a identificar e remover ruídos, a realizar o aguçamento de bordas, e a utilizar programação para executar operações matriciais essenciais no processamento de imagens.

Keywords—*OPENCV, python, aguçamento, filtragem, filtro laplaciano, imagem YUV, filtro rejeita notch, imagens no domínio da frequência*

I. INTRODUÇÃO

O relatório em questão tem o objetivo de discorrer sobre os desafios propostos pela disciplina de introdução ao processamento de imagens.

Os seguintes tópicos serão abordados:

- Expansão de pixels e interpolação de imagens;
- Aplicação de filtro laplaciano e filtro de suavização Gaussiano;
- Filtro rejeita-notch, mediante o uso de filtros passa-altas Butterworth.

A primeira questão aborda a extração de frame de um arquivo YUV, expansão da resolução e suavização dos pixels. Um arquivo YUV é um formato de arquivo que armazena dados de vídeo em um formato de separação de componentes de cor. O formato YUV separa a informação de luminância (Y) da informação de crominância (U e V). A luminância representa o brilho da imagem, sendo responsável pelo detalhamento da imagem em preto e branco. A crominância representa informação de cor, onde U indica a diferença entre o azul e a luminância e o V indica a diferença entre o vermelho e a luminância. O formato **RGB (Red, Green, Blue)** onde cada cor em uma imagem é definida pela intensidade desses três componentes são combinados para gerar a vastidão de cores existentes.

A segunda questão discute o processo de aguçamento (sharpening) de imagens, mediante o uso de um Filtro Laplaciano no domínio espacial e filtro de suavização Gaussiano. A aplicação do filtro Gaussiano à imagem original permite reduzir ruídos e detalhes finos, preparando-a para a detecção de bordas. O uso do filtro Laplaciano destaca as regiões de mudança rápida de intensidade, realçando bordas e detalhes finos. A combinação de ambos os filtros melhora a qualidade visual da imagem, tornando bordas e detalhes mais claros aos olhos humanos.

A terceira questão aborda os filtros passa-altas Butterworth são permitir a passagem de frequências acima de uma frequência específica e atenuando as frequências abaixo dela, o que ajuda a realçar detalhes de alta frequência na imagem. O filtro rejeita-notch possui o objetivo de remover padrões de interferência periódica, como o moiré, que aparecem em frequências específicas na imagem. A combinação de ambos os filtros permitem a remoção do ruído como será apresentado no decorrer do relatório.

II. METODOLOGIA

A. Parte 1 - Expansão de pixels e interpolação de imagens

Utilizando um arquivo YUV de exemplo, `foreman.yuv`, foi especificado a criação de uma função `LER_YUV`, que recebe os parâmetros:

- **src**: arquivo YUV;
- **largura**: largura da resolução do arquivo;
- **altura**: altura da resolução do arquivo;
- **frame**: frame selecionado do vídeo.

O vídeo possui resolução de 352x288 onde os componentes são disponibilizados no formato 4:2:0. Ou seja, os componentes U e V têm metade dos pixels em cada direção, ou seja, possuem resolução de 176x144.

Para chegar no frame passado como parâmetro da função, é necessário descobrir o offset do vídeo para capturar o frame. Como os componentes U e V têm resolução reduzida em 2 em ambas as direções (4:2:0), a largura e a altura são divididas por 2 antes de multiplicar. O tamanho total de um frame é a soma dos pixels da componente Y e duas vezes os pixels de uma das componentes U ou V (U e V têm o mesmo número de pixels). O offset é calculado multiplicando o número do frame pelo tamanho do frame. Este offset indica o início do frame desejado no arquivo YUV.

$$\text{offset} = \text{frame} \times (\text{largura} \times \text{altura} + 2 \times (\text{largura} // 2 \times \text{altura} // 2))$$

Com o frame em mãos, é necessário realizar a expansão dos pixels, para isso criamos uma matriz de 0's temporária onde nos índices pares colocaremos os valores do frame. Após a expansão, faremos a substituição dos valores dos 0's restantes trocando os pixels pretos pelo imediato superior ou imediato à esquerda. Quando possível escolher ambos sempre usar o superior.

Tabela 1 - Pixels originais

128	129	133
126	138	153
131	112	64

Tabela 2 - Pixels expandidos e substituídos

128	128	129	129	133	133
128	128	129	129	133	133
126	128	138	129	153	133
126	128	138	129	153	133
131	128	138	129	153	133

Para realizar o plot da imagem, é necessário a conversão de YUV para RGB. Contudo, como a resolução das componentes U e V são menores que Y, é necessário dobrar a largura e altura das componentes U e V.

Foi solicitado a pesquisa de uma maneira melhor para substituir os pixels pretos (0's) da imagem que foi dobrada. O método escolhido foi realizar a média dos pixels ao redor para obter um valor mais suave.

Tabela 3 - Pixels interpolados

128	128	129	129	133	133
128	128	129	129	133	133
126	126	138	138	153	153
131	131	112	112	64	64
131	131	112	122	64	64

Para colorir o frame foi utilizado uma função YUV420_to_RGB que realiza normalização dos dados conforme o padrão de conversão de cores, no final ela retorna a imagem colorida no padrão RGB possibilitando o plot.

B. Parte 2 - Aplicação de filtro laplaciano e filtro de suavização Gaussiano

A segunda parte do relatório realiza o processo de aguçamento (sharpening) de imagens, mediante o uso de um Filtro Laplaciano no domínio espacial. Foi realizada a convolução na imagem `Image1.png` utilizando dois kernels.

Tabela 4 - Kernel centro 4 negativo

0	1	0
1	-4	1
0	1	0

Tabela 4 - Kernel centro 8 negativo

1	1	1
1	-8	1
1	1	1

Para aplicar a suavização Gaussiana, foi utilizada a função `GaussianBlur` do `opencv` com desvio padrão do kernel gaussiano $\sigma=0,5$ e $\sigma=1,0$ para kernel com centro -4. Para aplicar a filtragem utilizando os kernels das tabelas 4 e 5, foi utilizada a função `filter2D` do `opencv`.

C. Parte 3 - Filtro rejeita-notch, mediante o uso de filtros passa-altas Butterwoth

Nesta terceira e última parte realizar o processo de filtro rejeita-notch, mediante o uso de filtros passa-altas Butterwoth (com $n=4$) na imagem disponibilizada `moire.tif` que possui o padrão de Moiré evidentes.

Inicialmente utilizamos a função `fft2()` do `numpy` para transformar a imagem para o domínio da frequência e reposicionar os quadrantes da imagem de forma que as baixas frequências sejam centralizadas e as altas frequências sejam deslocadas para as bordas e cantos, utilizando a função `fftshift()` do `numpy`.

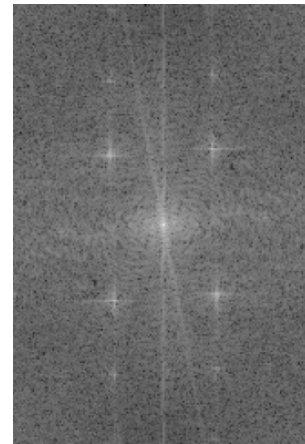


Figura 1 - Espectro de magnitude

A função `notch_reject_filter` cria um filtro rejeita-notch para uma imagem no domínio da frequência. Este filtro é usado para atenuar frequências específicas que causam interferências, como o efeito moiré.

Parâmetros da função:

- **linhas:** Número de linhas da imagem;
- **colunas:** Número de colunas da imagem;
- **d0:** Raio de corte do filtro, que define a distância máxima das frequências a serem rejeitadas;
- **uk, vk:** Coordenadas das frequências centrais a serem rejeitadas.

A fórmula matemática que descreve o cálculo realizado pela função pode ser entendida em termos dos cálculos das distâncias no espaço de frequências e a aplicação de uma condição de rejeição baseada nessas distâncias.

$$D_{uv} = \sqrt{\left(u - \frac{\text{linhas}}{2} + uk\right)^2 + \left(v - \frac{\text{colunas}}{2} + vk\right)^2}$$

$$D_{mu\nu} = \sqrt{\left(u - \frac{\text{linhas}}{2} - uk\right)^2 + \left(v - \frac{\text{colunas}}{2} - vk\right)^2}$$

Nas especificações foi solicitado o uso dos pares abaixo que foi aplicado na fórmula acima e na verificação da rejeição das frequências.

1. $d0 = 10, uk = 39, vk = 30$
2. $d0 = 10, uk = -39, vk = 30$
3. $d0 = 5, uk = 78, vk = 30$
4. $d0 = 5, uk = -78, vk = 30$

A condição de rejeição da frequência é dada por:

- Se $D_{uv} \leq d0$ ou $D_{muv} \leq d0$, o valor do filtro é definido como 0, indicando a rejeição dessas frequências.
- Caso contrário, o valor do filtro é definido como 1, indicando a passagem dessas frequências.

III. RESULTADOS

A. Parte 1 - Expansão de pixels e interpolação de imagens

Aplicando os métodos de expansão da resolução, ou seja dobrando o valor do frame e utilizando o método da substituição dos 0's, percebemos que na figura 2 existem linhas verticais na imagem.

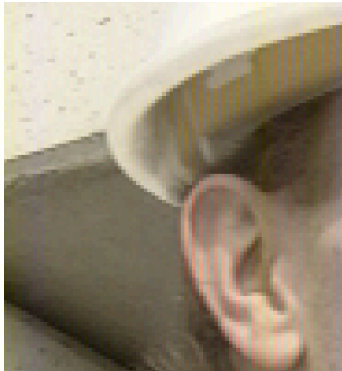


Figura 2 - Frame YUV expandido utilizando replicação

O mesmo não ocorre na figura 3 onde a mesma expansão foi realizada, mas o método de substituição dos zeros, o utilizado sendo cálculo de média.



Figura 3 - Frame YUV expandido utilizando cálculo de média

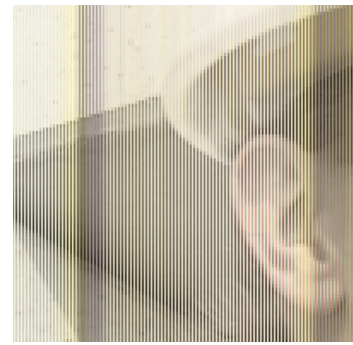


Figura 4 - Frame YUV expandido duas vezes utilizando replicação

Dada a implementação realizada e a forma de substituição dos pixels, observa-se que as linhas verticais ficam mais acentuadas na figura 4. Contudo, ao realizar a substituição seguindo a sequência de alterar todas as colunas e depois todas as linhas é possível obter um resultado mais satisfatório e sem a ocorrência das linhas verticais.

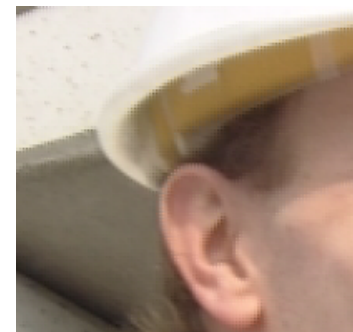


Figura 5 - Frame YUV expandido duas vezes utilizando cálculo de média

É possível observar o aumento de listras na figura 3, enquanto que na figura 5 ocorre a redução da qualidade da imagem.

B. Parte 2 - Aplicação de filtro laplaciano e filtro de suavização Gaussiano

A aplicação dos métodos discutidos na Seção II resultaram nas figuras 6, 7 e 8. Vale observar que obtive resultado inesperado ao utilizar kernels de centro positivo (exemplificado na tabela 5). A imagem final possui excesso de cor branca como observado na figura 9 o que difere da teoria estudada durante a disciplina onde pouca diferença ou nenhuma diferença era para ser observada entre o uso do kernel com centro positivo e negativo.



Figura 6 - Laplaciano centro 8 negativo



Figura 7 - Laplaciano centro 4 negativo $\sigma = 0,5$



Figura 8 - Laplaciano centro 4 negativo $\sigma = 1,0$

Tabela 5 - Kernel centro 8 negativo

-1	-1	-1
-1	8	-1
-1	-1	-1



Figura 9 - Laplaciano centro 8 positivo

Subjetivamente a figura 7 aparenta ser mais agradável de observar e analisar detalhes.

C. Parte 3 - Filtro rejeita-notch, mediante o uso de filtros passa-altas Butterworth

Na figura 10 é possível observar claramente o efeito Moiré na imagem. Após realizar o processo de filtragem, podemos observar o resultado na figura 11, contudo na borda ainda existem os resquícios do efeito.

O efeito moiré ainda é visível próximo das bordas da imagem, e isso ocorre porque não foi aplicado um padding na imagem no domínio da frequência antes de realizar a filtragem rejeita-notch. O padding é necessário para aumentar o tamanho da imagem e evitar que as bordas interfiram na filtragem, garantindo uma aplicação mais eficaz do filtro rejeita-notch.

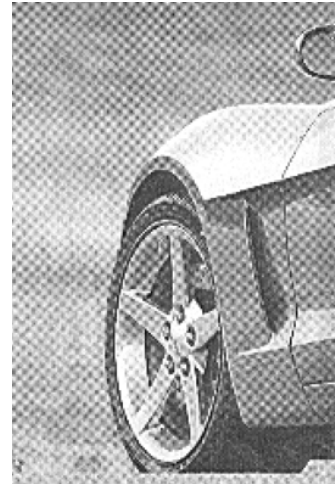


Figura 10 - moire.tif original



Figura 11 - Após remoção do efeito Moiré com filtro rejeita-notch

IV. CONCLUSÃO

A maior parte das atividades solicitadas foram realizadas com sucesso. Foi bastante proveitoso exercitar o processamento de imagens e aplicação do conteúdo apresentado durante a disciplina na realização das atividades.

REFERÊNCIAS

- [1] Slides da disciplina Introdução ao Processamento de Imagens ministrada pelo professor Bruno Macchiavello, Universidade de Brasília.
- [2] Imagens disponibilizadas pelo professor Bruno Macchiavello, Universidade de Brasília.