

INF394 - Processamento Digital de Imagens

Trabalho Prático - Parte 1

Prof. Marcos Henrique Fonseca Ribeiro

2018/I

1 Observações Gerais Sobre o Trabalho

1. Data do início da entrega do trabalho: **04/07/2018**, quarta-feira, às 18:00.
2. Data limite de entrega do trabalho: **11/07/2018**, quarta-feira, às 23:59.
3. Formato de entrega: arquivo **.zip**, contendo os arquivos pedidos ao longo do roteiro.
4. Meio de entrega: PVANet
5. Nesta parte, o trabalho pode ser desenvolvido individualmente ou em duplas.

2 Introdução

Utilizando a estrutura básica de programa com interface gráfica disponibilizada no PVANet ou a linguagem/tecnologia que preferir, desenvolva um software com funcionalidades que sejam capazes de cumprir as tarefas especificadas nas seções a seguir. Antes de prosseguir, no entanto, algumas observações **importantes**:

- Caso não vá utilizar uma linguagem multiplataforma, como Python ou Java, disponibilize juntamente com os arquivos entregues as instruções para instalação, compilação ou qualquer outra que for necessária para que uma pessoa consiga instalar e executar o produto final em Windows, Linux ou Mac.
- **Está vetado**, para este trabalho, o uso de qualquer biblioteca, módulo pronto, framework ou coisa que o equivalha que já implemente os algoritmos e técnicas de Processamento Digital de Imagens propriamente dito como, por exemplo, *OpenCV*. Você deve implementar estes recursos.
- **Está liberado** o uso de qualquer biblioteca, módulo, framework etc. que não seja para a finalidade descrita acima. Por exemplo, esteja à vontade para utilizar bibliotecas para manipulação de interface gráfica, manipulação de arranjos, matemática, gerenciamento de arquivos no SO etc.

- A exceção à regra acima está no uso de soluções para a carga de uma imagem a partir de um arquivo para um objeto, estrutura de dados ou afins; para a visualização de imagens e para o salvamento das mesmas. Sendo assim, por exemplo, pode-se utilizar o módulo Python PIL (*Python Image Library*) no trabalho para ler dados de arquivos, salvar arquivos, exibir imagens na tela. Porém não se poderá utilizar os recursos já presentes na mesma no que diz respeito ao processamento da imagem como, por exemplo, a conversão para uma imagem binária.

Pode-se utilizar como fonte de estudo o material disponibilizado no PVANet ou qualquer outro que desejar. Caso venha utilizar alguma solução diferente das vistas em sala de aula e fornecidas no material do professor, coloque a fonte (livro, site etc.) de onde foi tirada a solução.

Por fim, apesar de não ser obrigatório, recomenda-se a utilização da estrutura básica de software fornecida no site da disciplina, uma vez que a mesma já traz consigo exemplos e algumas funcionalidades de apoio, como desfazer operações, salvar com novo nome etc. Pode-se evoluir também a estrutura básica, caso deseje, acrescentando ou melhorando estas funcionalidades de apoio. Porém, para fins de avaliação, será considerada só a parte que disser respeito ao conteúdo da disciplina, isto é, processamento de imagens.

A Seção 3 traz uma breve explicação do que já se encontra implementado no esqueleto da estrutura básica fornecida. A especificação do trabalho propriamente dita se encontra na Seção 4. Porém, mesmo para os que não pretendem utilizar a estrutura fornecida, recomenda-se a leitura da próxima seção, pois a mesma traz explicações também de algumas premissas adotadas na concepção do trabalho.

3 Estrutura Básica Disponibilizada

Esta seção tem como objetivo fazer uma breve descrição dos recursos já implementados no esqueleto do software em Python, bem como fazer observações pontuais. Primeiramente, esta estrutura básica implementa sua interface gráfica (GUI) usando o pacote padrão do Python para esta finalidade, o *TkInter*. Existem alternativas a este pacote, com a mesma finalidade.

O pacote trabalha com o conceito de *widgets*, que são componentes de interface gráfica que o programador pode acrescentar à sua interface, bem como algumas janelas e caixas de diálogo padrão, como as de seleção de pastas e arquivos para abertura. Nos exemplos já implementados no esqueleto, encontram-se a demonstração da criação do uso de diversos desses *widgets*, porém, tenha em mente que diversos outros estão disponíveis. Para a documentação completa do pacote, visite: <https://wiki.python.org/moin/TkInter>.

A seguir, temos uma breve descrição sobre as funcionalidades que já se encontram implementadas:

- Menu Arquivo
 - Abrir. Realiza a abertura de uma imagem para a tela. Traz consigo exemplo de uso de caixas de diálogo para abertura de arquivos. Verifique que é possível filtrar os arquivos listados por tipo. Ao se abrir o arquivo, o conteúdo da imagem é carregado para um *widget*

do tipo *Canvas* (tela de pintura), que funciona basicamente como um contêiner da imagem.

- Salvar. Salva a imagem atualmente carregada na tela para o arquivo que originou a mesma. Atenção: esta operação sobrescreve o arquivo original. Traz consigo também exemplo de uso de tratamento de exceções e janelas de mensagem, caso se tente salvar um arquivo sem que haja uma imagem aberta.
- Salvar Como. Salva a imagem atualmente carregada na tela, porém permitindo selecionar um novo local e nome para a mesma.
- Fechar Imagem(ns). Limpa o *Canvas* e fecha as referências à(s) imagem(ns) abertas no momento.
- Sair. Encerra a execução do programa.

- Menu Editar

- Desfazer. Internamente, na implementação da GUI, são constantemente mantidas duas referências a objetos da classe *Imagen*. A imagem atual, de trabalho, que está exibida na tela, é referenciada no atributo *img* da GUI. Sempre antes de se realizar um processamento sobre esta imagem, uma referência para a mesma é salva no atributo *imgOld*. Após a operação de processamento, a nova imagem é referenciada no atributo padrão (*img*), mas uma cópia da versão anterior da foto permanece referenciada (*imgOld*). A operação Desfazer simplesmente troca as referências para estes dois objetos, fazendo com que a versão antiga retorne à referência de trabalho e seja exibida novamente na tela. Ao se evocar novamente esta funcionalidade, o efeito é trocar os objetos de referência novamente, fazendo com que a versão processada retorne à tela e à referência de trabalho.

- Menu Imagem

- Informações gerais. Exibe um pequeno sumário sobre o tipo e dimensões de imagem, bem como sua representação: RGB, L (monocromática), 1 (binária) etc. Também traz consigo um exemplo do uso de caixas de mensagens.
- Converter. Permite a conversão de um tipo de representação de imagens em outro, selecionado pelo usuário. Utiliza o recurso do pacote PIL específico para esta operação. Este tipo de uso de pacotes de processamento (conversão de tipos) de imagens está permitido para o aluno.

- Menu Cores

- Tons de cinza. Permite a conversão da imagem para uma escala de tons de cinza. No caso, a representação da imagem *de facto* ainda permanece RGB. O que se faz é calcular o nível de intensidade de cinza desejado e replicar este valor nas três componentes. Existem diversos métodos de conversão, e a referência a vários deles já se encontram neste menu, embora implementados mesmo tem-se apenas dois: *Average* e *Componente R*. Fará parte do trabalho a implementação de

apenas um dos métodos restantes, explicações sobre este método a ser implementado encontra-se mais adiante. Em termos de interface, traz consigo exemplo do uso de submenus, ou menus em cascata.

- Inverter. Calcula o negativo de uma imagem.
- Pseudo binária. Converte a imagem em uma representação “pseudo-binária”. Isto é, a representação *de facto* ainda é RGB, mas todos os pixels da imagem são convertidos para os tons preto ($[0, 0, 0]$) ou branco ($[255, 255, 255]$).
- Halftoning. Permite a conversão da imagem em (pseudo) binária, usando técnicas de meio-tom. Existem várias técnicas para tal tarefa. Porém este tópico não está contemplado no escopo deste trabalho e será dado futuramente na disciplina. Por ora, o recurso está disponível como funcionalidade de suporte, em alternativa ao método de binarização anterior.
- Balanço de Cores. Permite que se ajuste (percentualmente) os níveis de intensidade de cada componente RGB da imagem, individualmente. Em termos de interface gráfica, traz exemplos de barras de rolagem (*slides*) que podem ser utilizadas para definição de valores dentro de um intervalo arbitrário.

Sempre que necessário empregar a conversão para **tons de cinza**, use a técnica da Luminância (*Luminosity*), na qual utilizaremos a seguinte notação:

- r, g, b : componentes RGB de um pixel em um ponto (x, y) da imagem $f(x, y)$
- $g(x, y)$ é o pixel nas coordenadas (x, y) da imagem resultante de uma transformação sobre $f(x, y)$

E a transformação é dada pela obtenção de um valor $G = 0,299r + 0,587g + 0,114b$, que é aplicado à imagem transformada pela relação $g(x, y) = (G, G, G)$.

Por fim, vale ressaltar que a organização de menus e opções fornecidas são mero exemplo de uma possível organização para os mesmos e, além disso, algumas das opções sequer virão a ser implementadas no trabalho. Portanto, esteja livre para alterar essa organização de menus à vontade. A próxima seção traz as especificações das tarefas que devem ser cumpridas neste trabalho, bem como quaisquer explicações adicionais que sejam necessárias.

4 Tarefas

Cada tarefa a seguir pode necessitar da implementação de mais de uma técnica de PDI. Esteja atento à isso. Cabe ao aluno identificar o que é necessário para cumpri-la, dentro do conteúdo dado em sala de aula.

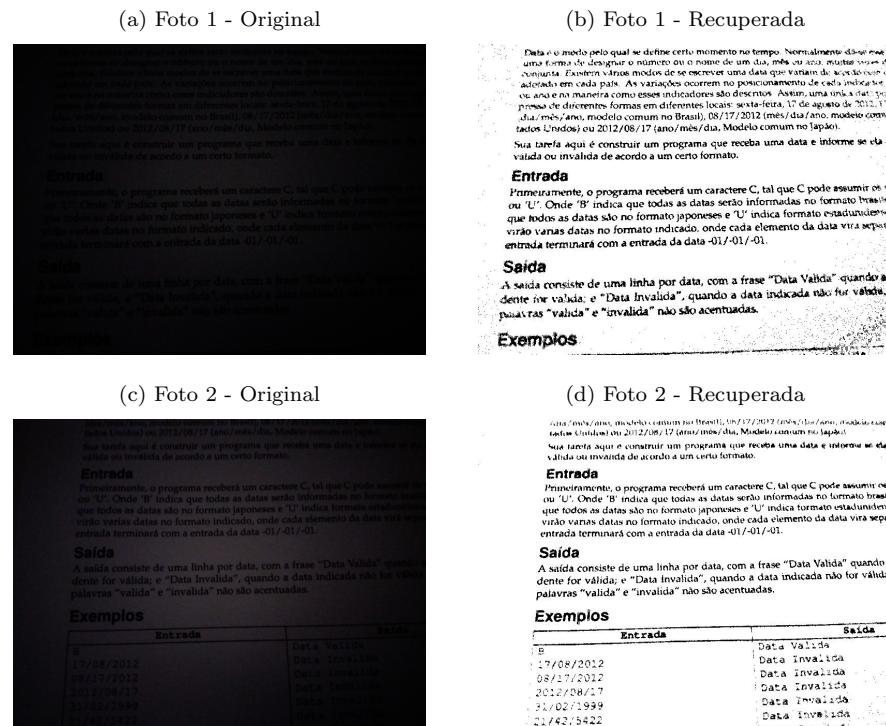
Importante: Descreva, num relatório, os passos desempenhados para se cumprir cada tarefa.

4.1 Melhoria de Imagens

Na subpasta “tarefa1”, encontram-se duas fotografias de um texto em papel, adquiridas sob condições muito ruins de iluminação. Você deve recuperar estas imagens para que as mesmas fiquem mais legíveis, utilizando **minimamente** técnicas para melhorar contraste, conversão para tons de cinza e limiarização (escolha um método de limiarização, dentre os apresentados em sala). É permitido o uso de técnicas adicionais. A escolha das técnicas é de responsabilidade dos alunos. A saída deve ser uma imagem pseudo-binária, isto é, uma imagem no padrão RGB, porém que contenha somente pixels pretos (0, 0, 0) e brancos (255, 255, 255).

A Figura 1 traz um exemplo de solução da tarefa para os arquivos contidos na referida pasta.

Figura 1: Recuperação de Imagens Obtidas sob Máis Condições de Luminosidade



4.2 Geração de Ruídos

Implemente uma funcionalidade que produza ruídos sobre uma imagem. O ruído deve ser do tipo “Sal e Pimenta”. O usuário deve fornecer dois parâmetros: o percentual de pixels da imagem afetado pelo ruído e a proporção entre sal e pimenta. Isto é, quantos pixels da imagem serão substituídos por um pixel de ruído (sal: branco, pimenta: preto) e, dentro deste conjunto de pixels, quantos porcento serão de sal e quantos serão de pimenta. O ruído deve ser distribuído em pixels aleatórios da imagem. A Figura 2 traz dois exemplos da aplicação do ruído sobre uma imagem, que encontra-se disponível na pasta “tarefa2”.

Figura 2: Ruído Sal e Pimenta



4.3 Restauração de Imagens - 1

Usando o recurso desenvolvido na tarefa anterior, a partir de uma mesma imagem original, produza 2 imagens com ruído sal e pimenta, em uma quantidade razoável de pixels, com a distribuição entre sal e pimenta da forma que desejar. Em seguida, utilize o seguinte método para restaurar aproximadamente a imagem original.

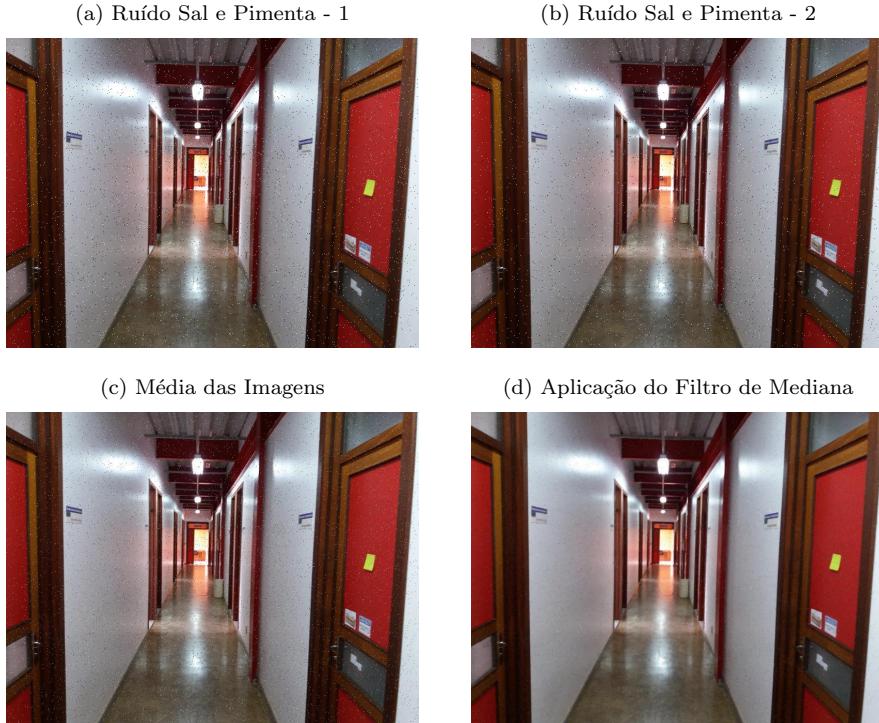
- Produza uma imagem cujos valores dos pixels são a média dos pixels correspondentes nas duas imagens ruidosas:

$$h(x, y) = \frac{f(x, y) + g(x, y)}{2}$$

- Aplique um filtro de mediana sobre a imagem resultante do passo anterior.

A Figura 3 traz um exemplo de aplicação da técnica descrita.

Figura 3: Restauração de Imagens Ruidosas



4.4 Restauração de Imagens - 2

Produza, usando o recurso desenvolvido na Seção 4.2, n imagens ruidosas a partir de uma mesma imagem original. Em seguida, restaure aproximadamente esta imagem produzindo uma nova imagem cujo valor seja a média de todas imagens ruidosas:

$$g(x, y) = \frac{\sum_{i=1}^n f_i(x, y)}{n}$$

Teste o resultado com diferentes valores para $n \in [2, 50]$. Compare seus resultados com a imagem original e veja se estão satisfatórios. Inclua estes resultados no seu relatório.

4.5 Composição de Imagem

Na pasta “tarefa5”, encontram-se três imagens monocromáticas. Cada uma contém as informações de uma das componentes RGB de uma imagem original, que deve ser recriada pelos alunos. Para tal, capture a informação contida em

cada uma delas e as utilizem para compor uma imagem colorida, em RGB. Após isso, produza uma imagem monocromática “realista” do papagaio, usando a técnica descrita no início do documento. A Figura 4 ilustra o resultado esperado.

Figura 4: Composição de Imagem a Partir de suas Componentes



4.6 Segmentação de Imagens

Implemente o filtro de detecção de linhas de **Sobel**. Muitas vezes, utiliza-se o resultado monocromático e negativo do filtro para finalidades práticas. Crie imagens usando estes recursos. As figuras 5 e 6 trazem exemplos desta tarefa.

Figura 5: Detecção de Descontinuidades - Filtro de Sobel

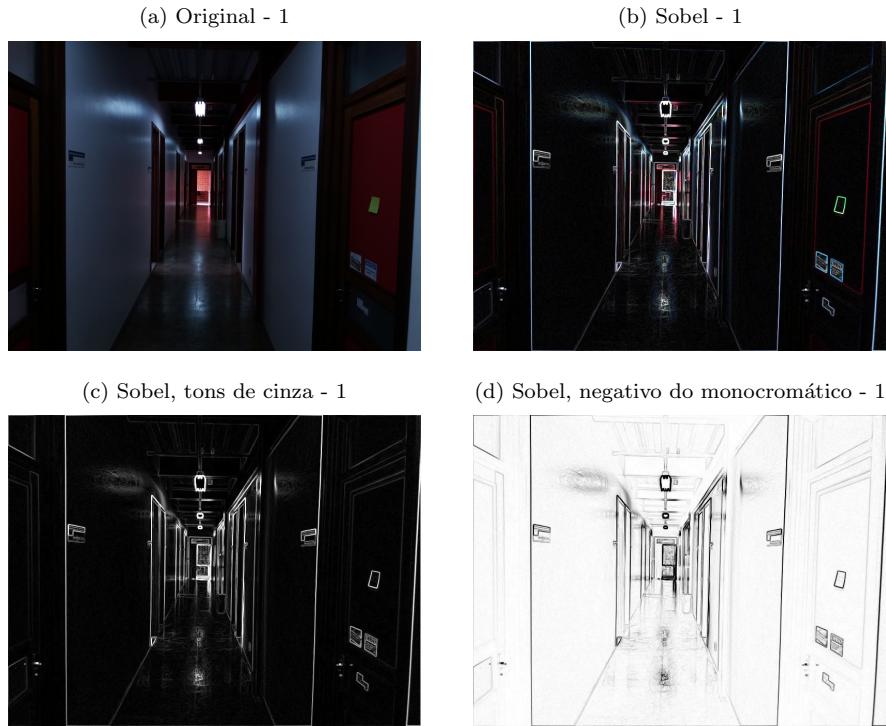
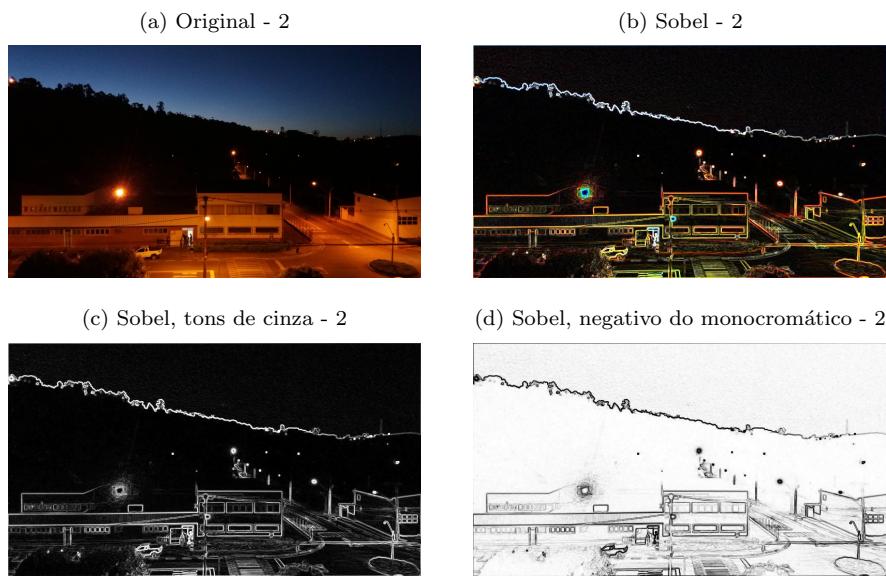


Figura 6: Detecção de Descontinuidades - Filtro de Sobel



4.7 Transferência de Estilo

Analisando o material didático disponibilizado, identifique e implemente uma técnica que possibilite fazer uma espécie de “transferência de estilo” de uma imagem para outra. Para testar, utilize as imagens para obter os resultados como mostrados nas figuras 7, 8, 9 e 10, a seguir.

Figura 7: Transferência de Estilo - Exemplo 1

(a) Foto base (1)



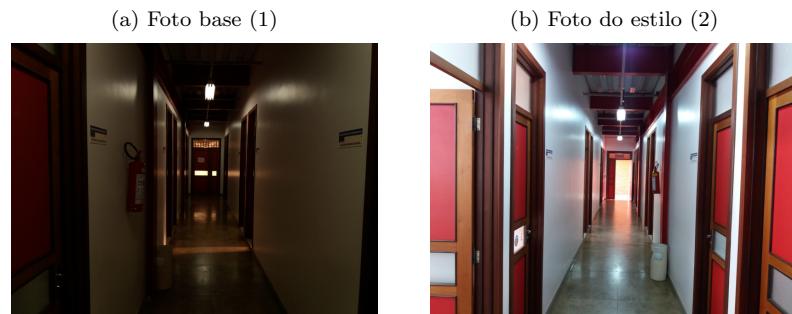
(b) Foto do estilo (2)



(c) Aplicação do estilo de (2) em (1)



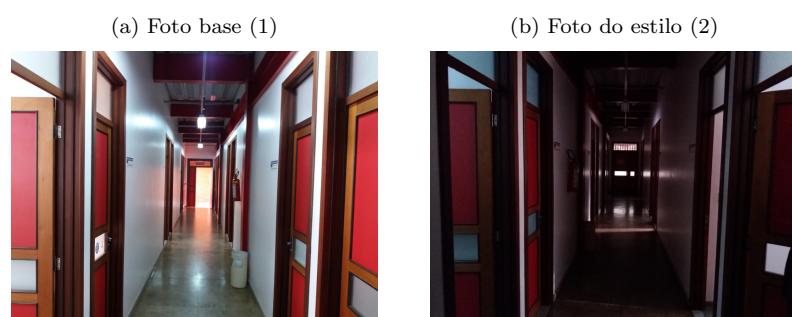
Figura 8: Transferência de Estilo - Exemplo 2



(c) Aplicação do estilo de (2) em (1)



Figura 9: Transferência de Estilo - Exemplo 3



(c) Aplicação do estilo de (2) em (1)

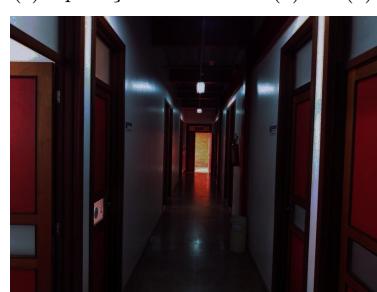


Figura 10: Transferência de Estilo - Exemplo 4



Note pela posição dos objetos que, embora similares, as fotos são diferentes e, de fato, o “estilo” de uma foi transferido para a outra. Todas imagens base e de estilo estão na pasta “tarefa7”.

4.8 Combinação Linear com Coeficientes Variáveis

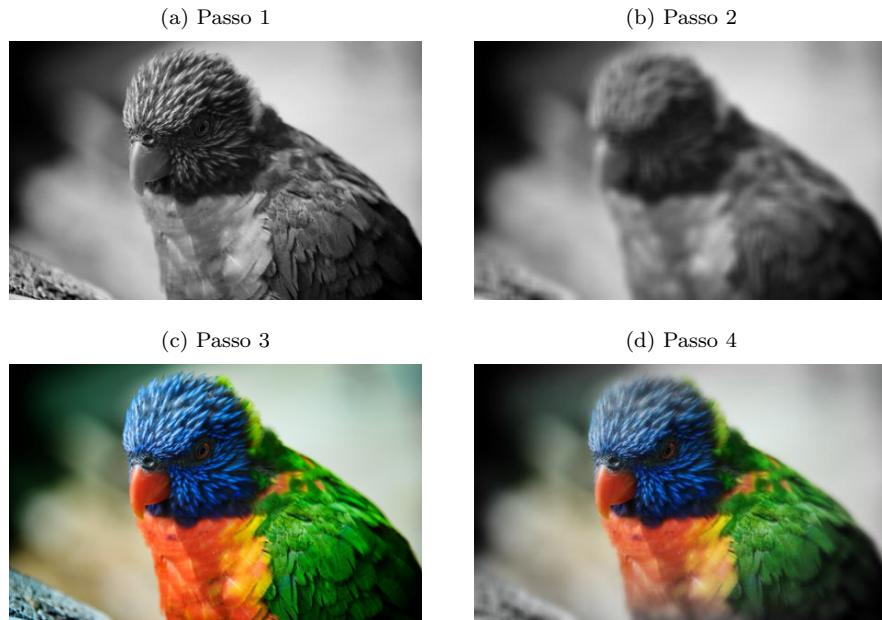
Na aula de “Operações Ponto a Ponto”, são mostradas técnicas de se combinar duas imagens, onde os coeficientes das combinações lineares produzidas variam no espaço. Implemente duas das combinações ilustradas. A primeira **deve** ser a variação quadrática em ambos eixos, com vértice no centro da imagem. A segunda, pode ser de escolha dos alunos.

Para ilustrar (e testar) a implementação da primeira combinação (quadrática), realize os seguintes passos.

1. Utilize a imagem monocromática do papagaio, produzida na tarefa 4.5
2. Aplique sobre a mesma um filtro de suavização da **média**, para gerar uma imagem razoavelmente borrada
3. Utilize a imagem colorida do papagaio, produzida na tarefa 4.5
4. Combine ambas imagens com a variação quadrática dos coeficientes para produzir um efeito similar ao mostrado a seguir

A Figura 11 ilustra os passos descritos acima.

Figura 11: Combinação Linear com Coeficientes Variáveis



As Figuras 12 e 13 trazem exemplos de combinações que podem ser utilizadas na sua segunda implementação. São exemplos meramente ilustrativos e não têm necessariamente que ser feitos da forma que estão aqui. Os alunos têm liberdade para decidir quanto a esta parte. As imagens destes exemplos estão disponíveis na pasta “tarefa8”.

Figura 12: Combinação Linear com Coeficientes Variáveis e Função Sigmóide

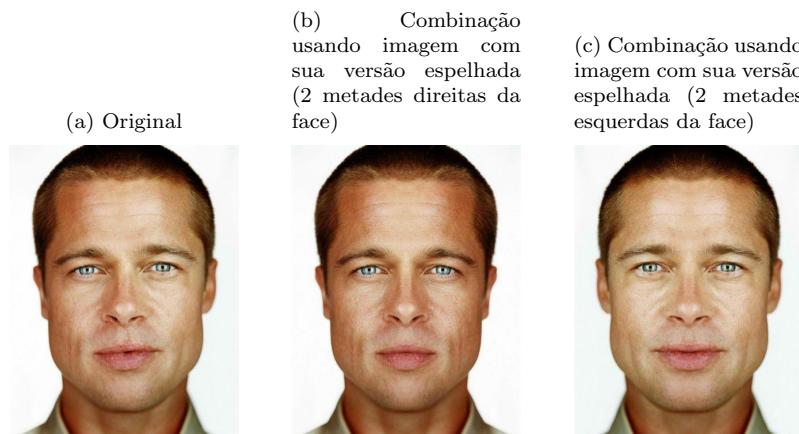
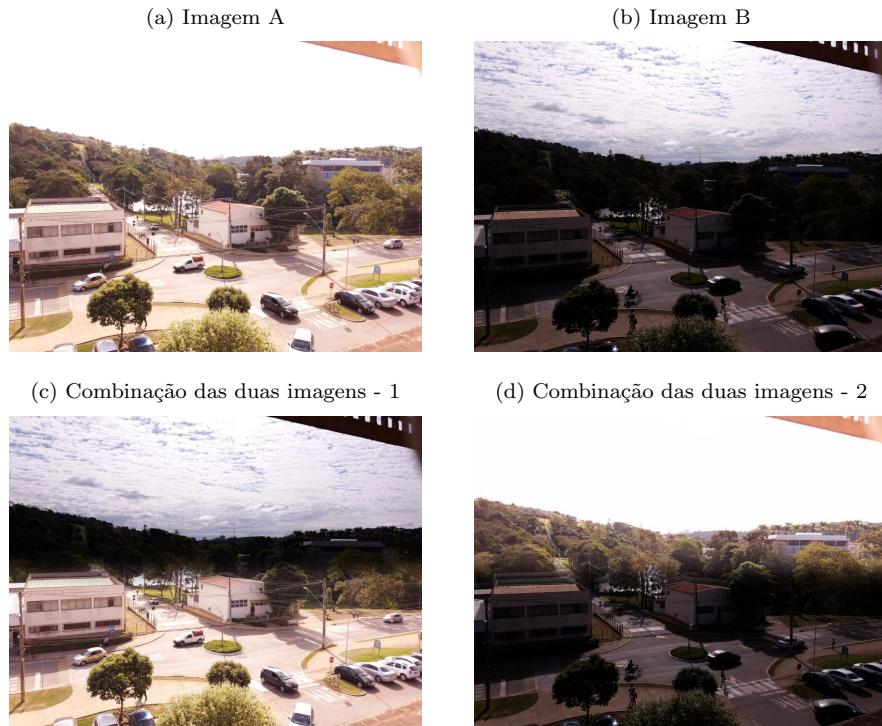


Figura 13: Combinação Linear com Coeficientes Variáveis e Função Sigmoidal



4.9 Realce

Implemente o filtro de realce laplaciano, usando tanto a vizinhança-4 quanto a vizinhança-8, conforme ilustrado nas aulas sobre filtragem espacial. A Figura 14 ilustra a aplicação dos filtros de realce sobre a imagem colorida do papagaio, produzida na tarefa 4.5.

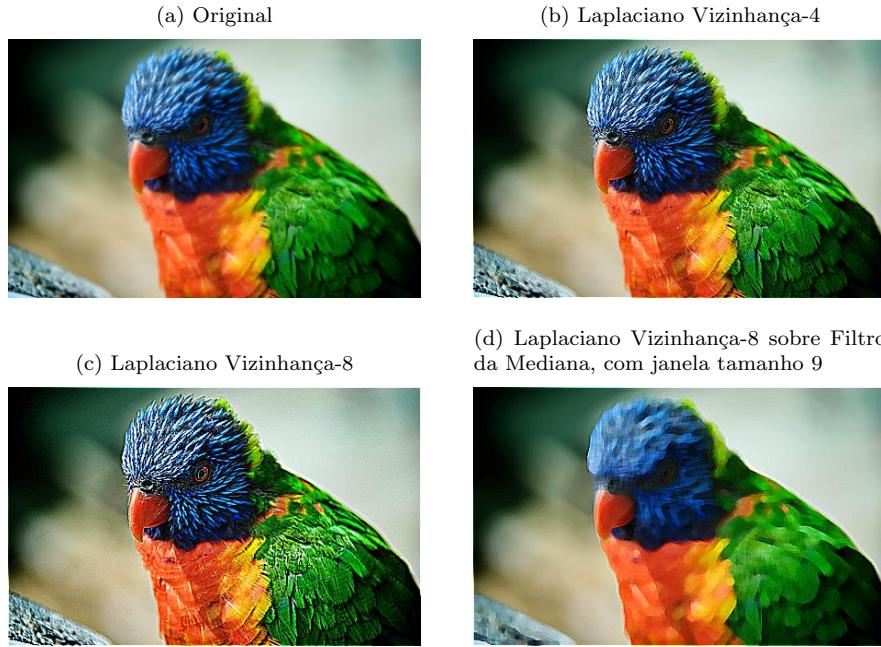
Por fim, aplique sobre uma imagem de sua escolha os filtros de mediana e o de realce em sequência, para produzir um efeito “artístico” que se assemelha a de pintura sobre tela, como mostrado na última imagem da Figura 14.

5 Entrega

O arquivo compactado *.zip* a ser enviado deve conter:

- Todos os arquivos *.py* (ou na linguagem escolhida) resultantes da versão final do programa desenvolvido pelo grupo.
- Arquivo de texto (*.pdf*) com um **relatório**, que deve:
 - Identificar o(s) aluno(s) que realizou(aram) o trabalho, com nome e matrícula
 - Conter instruções para instalação e execução do software desenvolvido como, por exemplo, quais pacotes necessitam estar instalados para o funcionamento, como compilar etc.

Figura 14: Realce



- Breve explicação a respeito de cada item implementado, descrevendo, em linhas gerais, como o método foi traduzido para a linguagem em questão, quais otimizações foram feitas, quais simplificações (se for o caso) foram feitas, decisões de projeto tomadas etc. Lembre-se de incluir as fontes utilizadas, caso lide com algo que fuja ao material dado em sala.
- Coloque também exemplos com imagens antes e depois de cada recurso implementado.
- Arquivos das imagens originais (porém em tamanho não muito grande) utilizadas pelo grupo para exemplificar os recursos no relatório, caso venham a utilizar imagens além das fornecidas pelo professor.

Bons estudos.
Prof. Marcos