

Processamento Digital de Imagens

Trabalho 1

Mariana Mendanha da Cruz
Engenharia de Controle e Automação
Faculdade de Tecnologia - UNB
Matrícula: 160136784
Email: mariana.mendanha.mm@gmail.com

Abstract—Trabalho de Processamento Digital de imagens utilizando Octave.

I. INTRODUÇÃO

O objetivo deste trabalho é abordar o tema Processamento Digital de imagens, mais especificamente o realce no domínio espacial e filtragem no domínio da frequência contando com a ferramenta Matlab/Octave e também por em prática os conhecimentos passados em aula na disciplina de Introdução a Processamento de imagens.

MATLAB é um software interativo de alta performance voltado para o cálculo numérico. Ele integra análise numérica, cálculo com matrizes, processamento de sinais e construção de gráficos em ambiente simples. Utilizando-se do pacote Image Processing Toolbox do Matlab/Octave é possível ler imagens como matrizes e com isso aplicar diversas operações, manipulando a imagem.

Com relação ao tema, é importante abordar alguns conceitos, sendo eles: Matrizes, convolução, interpolação, histograma.

Uma matriz é uma tabela retangular composta por escalares, onde chamamos estes escalares de entradas ou elementos. Tais elementos são organizados pela sua posição e seus valores podem conter significados. É de suma importância ressaltar para esse trabalho que matrizes podem ser manipuladas com operações matemáticas como dito anteriormente.

Convolução é simplesmente uma operação de somatório do produto entre duas funções, ao longo da região em que elas se sobrepõem, em razão do deslocamento existente entre elas. Quando falamos de utilizar a convolução em processamento de imagens trata-se de uma filtragem para extração de informações de interesse. Mais especificamente, o uso de filtros espaciais lineares é feito através de matrizes denominadas máscaras ou kernels (como são mais conhecidos na prática).

Interpolar é chegar a um resultado desconhecido através de fatores conhecidos. No caso de processamento de imagens seria criar novos pixels através dos dados dos pixels existentes.

Histograma para processamento digital é um diagrama constituído de um intervalo de níveis de cinza e sobre ele os valores correspondentes da imagem em questão representados por barras verticais.

II. METODOLOGIA

Nessa seção será abordada a resolução dos Problemas:

A. Problema 1.1

Foi pedido a criação de uma função chamada `dec_int` que dada uma imagem, diminua a imagem por um fator de N (múltiplo de 2) e depois interpole a imagem para o seu tamanho original, repetindo o valor do pixel mais próximo sem o uso de funções prontas do Matlab/Octave.

Então, criou-se uma função que recebe a matriz de uma imagem (Matriz 0) e um fator N . Dimensionou-se tal matriz recebendo os valores nas variáveis linha e coluna, cria-se uma nova matriz (Matriz 1) com valores $(x,y,z)(1:coluna,1:linha,1:formato)$ que será manipulada, divide-se então os valores das dimensões pelo fator N dado, criou-se vetores linha com tais valores com elementos espaçados linearmente entre 1 e os valores iniciais de linha e coluna, juntou-se esses vetores em um vetor maior com o auxílio de `meshgrid()` (Matriz 2). Finalmente, utilizando a função `interp3()` interpolando a referência Matriz 1+Matriz 0 nas coordenadas da Matriz 2 para obter a nova matriz.

Não terminando por aí, é feito o processo inverso para voltar a imagem ao seu tamanho original com os mesmos processos utilizados anteriormente, porém multiplicando as dimensões pelo fator N . Obtendo-se assim a imagem final.

B. Problema 1.2

Foi pedido a criação de uma função chamada `edge_improv` que utilize um filtro de aguçamento no domínio espacial a sua escolha de forma de melhorar a qualidade subjetiva de uma determinada imagem.

Então, criou-se uma função que recebe a matriz de uma imagem, utilizando o operador laplaciano de centro de

máscara negativo, negando-o e aplicando-o como filtro na imagem original, obtendo-se uma nova imagem que possui os detalhes finos da imagem original. Finalmente, subtraiu-se a imagem original pela imagem obtida chegando-se a imagem final aguçada.

C. Problema 1.3

Foi pedido a criação de um programa principal chamado Prog 1 que chama a função `dec_int` para a imagem 'test80.jpg'. Mostrar o resultado. Fazer um processo similar utilizando redução e interpolação bicúbica (`imresize`). Depois utilizar a função `edge_improv` para melhorar subjetivamente ambas as imagens interpoladas.

Então criou-se um programa principal (Prog1) com os diretórios das funções criadas `dec_int()` e `edge_improv()` explicadas anteriormente, recebe-se a matriz de tal imagem e mais o `input()` do usuário para o valor do fator N, em seguida, utiliza-se as função `dec_int()` e `edge_improv()` na imagem original mostrando seus resultados, em seguida, aplica-se `resize()` a imagem, com o mesmo fator N dado inicialmente, novamente aplica-se `resize()` para obtenção da imagem com dimensões iguais as da original e por último, utiliza-se `edge_improv()` para a obtenção das imagens resultantes, mostrando-as.

D. Problema 1.4

Foi pedido uma forma de obter-se um resultado melhor para a interpolação de uma imagem reduzida ao seu tamanho original.[Resolução apenas teórica].

Quando é aplicado um "zoom" em uma imagem, isto é a ampliação de uma imagem, perde-se informações, pois a quantidade de pixels de uma imagem e suas dimensões são constantes como pode ser visto na imagem a seguir:

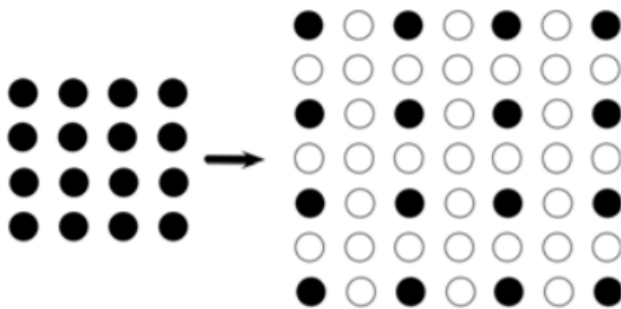


Fig. 1. Imagem ampliada

Para resolver este problema pode ser aplicada a interpolação, que nada mais é que o preenchimento desses espaços. Para que a imagem permaneça a mesma, a escolha desses pixels de preenchimento devem ser escolhidos de acordo com um método, para o problema 1.1 utilizou-se a

interpolação por vizinho mais próximo, tal método consiste na interpolação mais simples de ser implementada, contudo, apresenta desvantagens por causar distorções em detalhes finos e frequentemente apresenta o efeito jaggie (introduz serrilhamento nas bordas).

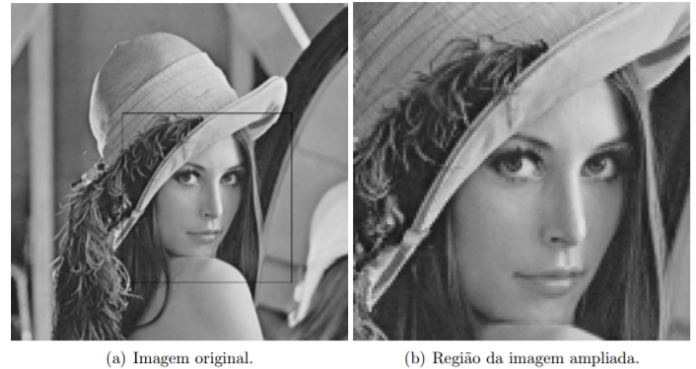


Fig. 2. Imagem ampliada

Para melhorar tais efeitos podemos utilizar um novo tipo de interpolação, como por exemplo a Interpolação Bicúbica, ela utiliza a média ponderada dos pixels originais que se encontram ao redor do novo pixel, utilizando dezesseis pixels vizinhos. É atribuído a cada um o peso correspondente à sua distância geométrica ao novo pixel e, por meio do uso de pesos baseados na spline cúbica, é calculado o seu valor de intensidade. A Figura abaixo mostra o esquema da interpolação bicúbica:

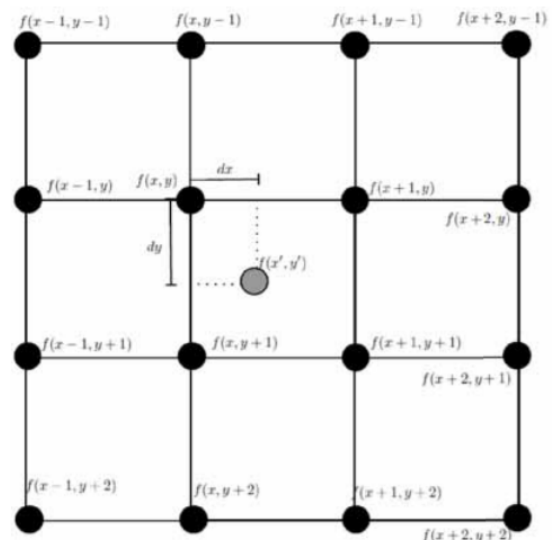


Fig. 3. Interpolação Bicúbica

O método da interpolação bicúbica não causa o serrilhamento tão exagerado como causado pelo vizinho mais próximo e também não suaviza tanto a imagem quanto

uma interpolação bilinear. É frequentemente utilizado em softwares de edições de imagens.



Fig. 4. Imagem ampliada

Existem outras técnicas de interpolação que inclusive são melhores, como a Interpolação com base na Convolução, porém não será discutido.

E. Problema 2.1

Pediu-se para testar o realce por correção gamma para três imagens diferentes, estas: "car.png", "crowd.png" e "university.png". Com pelo menos 2 valores de gamma para cada imagem e assim, mostrar o melhor resultado para cada uma.

Leu-se as imagens, testou-se por meio de laços diversos valores de gamma com cada imagem, selecionou-se, então 3 valores para amostragem e escolheu-se cada uma das imagens com melhores resultados, mostrando-as.

F. Problema 2.2

Pediu-se para equalizar as três imagens diferentes, estas: "car.png", "crowd.png" e "university.png". Mostrar seus resultados e de uma delas mostrar o histograma e CDF antes e depois da equalização.

Leu-se as imagens, equalizou-as por meio da função histeq(), mostrou-se cada uma delas, em seguida, calculou-se os valores de CDF da imagem "car.png" original e equalizada e com tais valores foi possível fazer a plotagem de um gráfico da função com um intervalo de 0 a 255 (por conta dos níveis de cinza).

G. Problema 2.2

Pediu-se para comentar os resultados indicando o melhor método para cada imagem, essa questão será resolvida mais a frente na seção de Resultados.

III. RESULTADOS

A. Problema 1

Visto anteriormente como foram resolvidos os problemas apresentados, vamos aos resultados:

Iniciou-se com a imagem original:



Fig. 5. Imagem original

Utilizando o fator de redução $N=2$ (dimensão reduzida pela metade) temos a imagem:



Fig. 6. Imagem reduzida pela metade

Os pixels da imagem ficam bem mais visíveis, com efeito aliasing, qualidade pior.

Utilizando o fator de redução $N=1/2$ (dimensão ampliada para a original) temos a imagem:



Fig. 7. Imagem ampliada pós redução

Os pixels da imagem ainda estão mais visíveis que a original porém mais suaves que a imagem reduzida, a qualidade não volta a ser o que era na original.

Após o processo de ampliação, teremos a imagem resultante com aguçamento, que pode ser vista a seguir:



Fig. 8. Imagem aguçada após ampliação

Podemos visualizar que o aguçamento ajudou muito tirando o borrado da imagem anterior, melhorando assim a definição da imagem, Mas ainda sim, não chega a ser melhor que a original.

Agora vamos comparar resultados de todo o processo apresentado entre as funções que foram implementadas e o método com a função `resize()`:

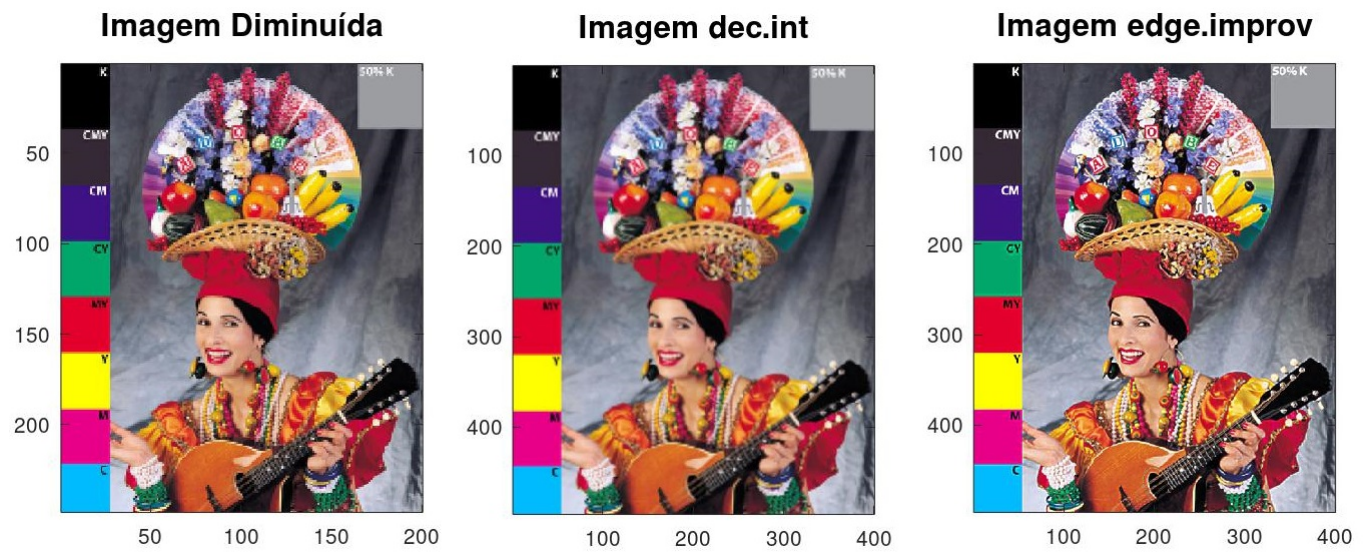


Fig. 9. Processo com implementação de funções

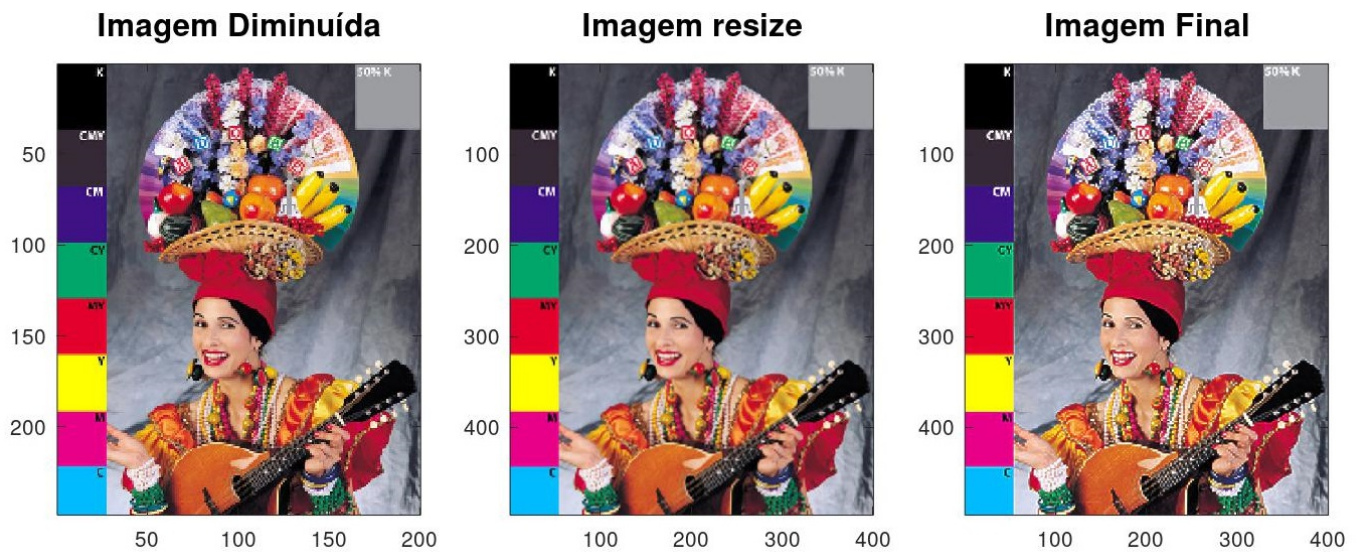


Fig. 10. Processo com `resize()`

Conseguimos perceber que a diferença entre as imagens reduzidas e as ampliadas é extremamente sutil, sendo que a imagem `dec.int` aparenta ter bordas mais suaves que a `resize`. Mas também, pode-se observar que as imagens finais após o aguçamento tem diferenças perceptíveis, como pode ser notado que a imagem final do método `resize` possui efeito maior de jaggies e distorções, portanto, a imagem final do processo com implementação de funções é mais nítida, e então, melhor.

B. Problema 2

Visto anteriormente as resoluções dos problemas apresentados, vamos aos resultados:
Iniciou-se com as imagens originais:



Fig. 11. Imagens originais "car.png", "crowd.png" e "university.png".

Aplicando-se a correção gamma nas imagens, os melhores resultados obtidos foram:



Fig. 12. Imagens com correção gamma

Nota-se que as três imagens melhoraram, especialmente as imagens 2 e 3 que passaram de imagens escuras e sem visibilidade para imagens com bastante visibilidade. Como pode ser visto escolhe-se valores maiores que 1 de gamma para escurecer regiões escuras, enquanto que escolhe-se valores menores que 1 quando queremos clarear regiões escuras, que foi o caso das imagens 2 e 3. Outros testes com gamma podem ser vistos com mais detalhes no código do trabalho.

Em seguida, equalizou-se as imagens originais, obtendo-se as resultantes:



Fig. 13. Imagens equalizadas

Nota-se que a melhora na imagem 1 foi tremenda, seus níveis de cinza ficaram mais equilibrados que a imagem original, enquanto que, ouve uma piora na qualidade das imagens 2 e 3, sendo que a imagem 2 ficou estourada e a 3 ficou extremamente pixelada.

Enfim, plotou-se os histogramas e CDFs da imagem 1 antes e depois de sua equalização:

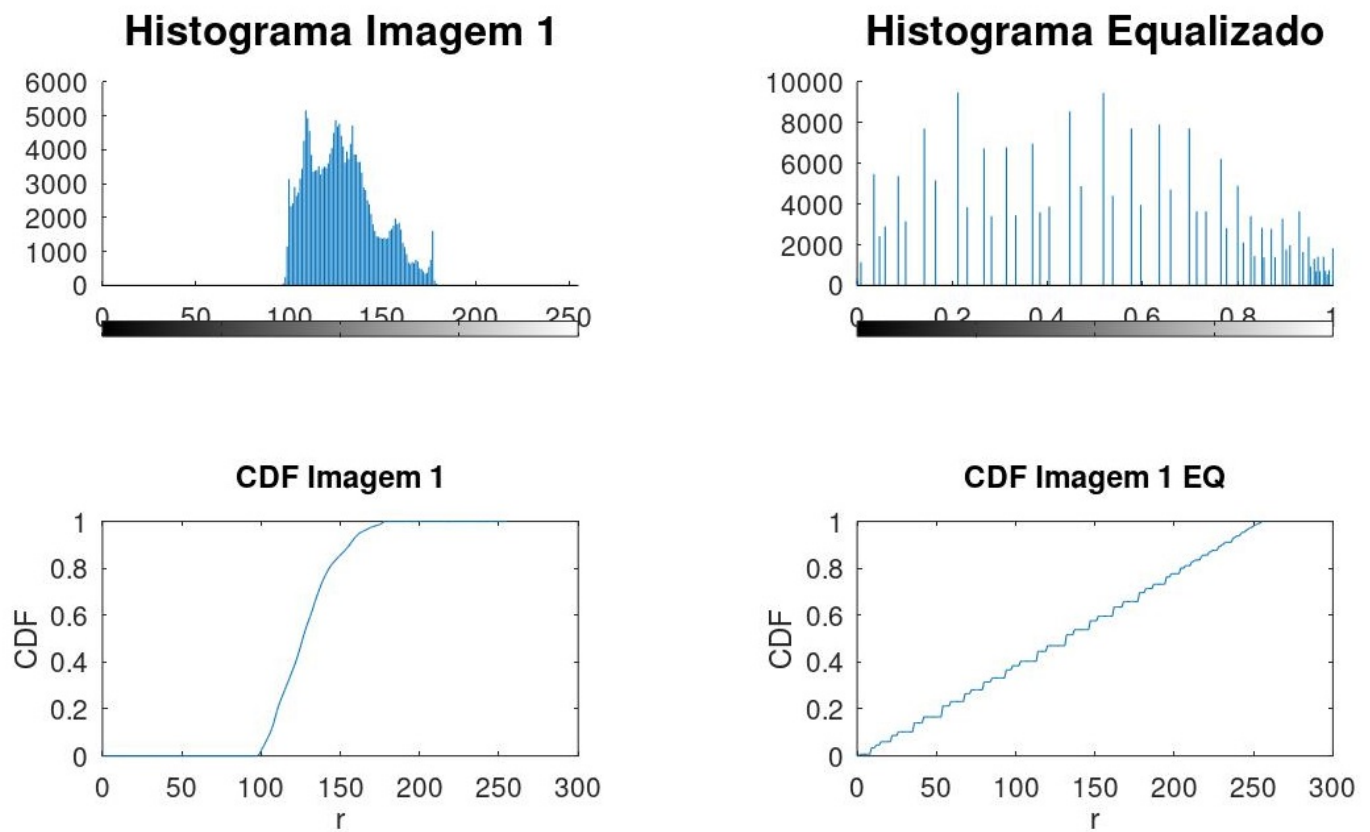


Fig. 14. Diagramas da imagem 1

Como pode ser visto, os níveis de cinza da imagem 1 se encontram muito mais espalhados no histograma após a equalização, com isso concluímos o que já pôde ser visto na própria imagem e o CDF reflete o próprio histograma, com uma distribuição mais equilibrada, próxima de uma reta diagonal.

IV. CONCLUSÃO

Podemos concluir no problema 1 que colocando lado a lado as imagens aguçadas com a original, têm-se que a imagem que passou pelos processos sem a função `resize()` é a mais próxima da original, mas mesmo próxima, a original ainda tem melhor visibilidade



Fig. 15. Comparação final

Podemos concluir no problema 2 que os melhores resultados obtidos com os realces de imagem são os seguintes:



Fig. 16. Melhores resultados

Em que a imagem 1 foi equalizada e as imagens 2 e 3 passaram por correção gamma com os valores mostrados anteriormente.

Por último, podemos dizer que o objetivo do trabalho foi alcançado, os temas propostos foram trabalhados, praticou-se os conceitos e conhecimentos abordados em aula e encontrou-se todos os resultados requisitados.

REFERENCES

Roteiro do trabalho.
Slides de Introdução a Processamento de imagens.
Livro texto - Digital Image Processing.
<https://viceri.com.br>
<https://infoescola.com.br>
<https://tecnoblog.net>
<https://stackoverflow.com>
<https://repositorio.unesp.br/bitstream/handle/11449/115892/000806982.pdf?sequence=1>
<https://octave.sourceforge.io>
<https://mathworks.com>
https://www.bogotobogo.com/Matlab/Matlab_Tutorial_Digital_Image_Processing_I.php