

Relatório Trabalho 1 – IPI

Gustavo Pereira Chaves
Departamento de Ciência da Computação
Universidade de Brasília
Brasília, Brasil
gustavopchaves@hotmail.com

Keywords— *imagem, processamento de imagem, interpolação, redução, histograma, CDF*

I RESUMO

Os experimentos aqui demonstrados visam realizar transformações no domínio espacial das imagens, realizando comparações entre diferentes métodos, de forma a validar a sua eficiência.

A respeito dos exercícios, o primeiro requer a construção de uma função que possibilita redução de uma imagem e posteriormente a sua interpolação, comparando o resultado com funções existentes. Já o segundo busca testar a transformação power-law para imagens com níveis de brilho distintos, buscando identificar o melhor valor gamma, além de contrapô-los as imagens geradas através do método de equalização de histograma.

Os códigos foram todos desenvolvidos em Python, utilizando as bibliotecas OpenCV, Numpy e Matplotlib, havendo sido testados em todas as imagens propostas fornecidas.

II INTRODUÇÃO

Este trabalho teve por finalidade a aplicação dos conhecimentos obtidos em sala de aula com relação a redimensionamento e interpolação de imagens, bem como testar filtros de realce e obter os melhores resultados possíveis dentro dos limites do experimento.

Inicialmente, é preciso discutir os conceitos de redução e interpolação. A redução diz respeito a diminuir o número de pixels de uma imagem, portanto, reduzindo suas dimensões. Esse método acaba provocando certa perda de dados, tendo em vista que alguns pixels são descartados durante o processo. Referindo-se a interpolação (aqui utilizada como retorno da imagem as dimensões originais), é o processo de aumentar o número de pixels de uma foto, ou seja, sua resolução, através da geração artificial deles. Tal processo, apesar de tornar a imagem maior, não necessariamente a tornará de melhor visualização. Tendo em vista que no processo de redução houve descarte de pixels, o processo de interpolação não fará a imagem retornar completamente ao seu estado original, já que houve perda de informação na realização das duas transformações.

Em segunda instância necessita-se entender a transformação power-law (ou Gamma). Esta, por sua vez, visa aumentar o contraste de imagens nas regiões de maior ou

menor brilho. Isso ocorre devido a curva realizada pela operação de exponenciação, que aumenta a inclinação mais rapidamente em função da variável x . Assim, o chamado gamma é o expoente ao qual cada pixel da imagem será submetido, de forma a alterar o seu brilho. Para valores entre 0 e 1, a função tende a aumentar o contraste nas regiões mais escuras, ou seja, ocorre um aumento no brilho final. Para valores maiores que 1 ocorre o efeito contrário, há um aumento do contraste nas zonas mais claras, que acarreta na diminuição do brilho final.

Por fim, é adequado descrever sobre os conceitos de histograma e CDF de uma imagem. Esse conceito torna-se necessário a partir do momento que o computador precisa interpretar uma imagem e compreender em que condições a mesma será apresentada para o olho humano. Para isso, deve-se contar quantas vezes cada nível de brilho ocorre em todos os pixels da imagem, e, ao fim, pode-se aferir se a mesma está muito clara, muito escura, ou com pouca distribuição (baixo contraste). A CDF é apenas a função de distribuição acumulada para cada nível da escala, ou seja, é a divisão dos pixels contados pelo total acrescidos da probabilidade anterior. Dessa forma, é possível equalizar o histograma da imagem, e distribuir esses pixels de forma mais uniforme pelos níveis de cinza.

De posse dessas informações, pode-se dar continuidade a exposição do experimento. Na seção III (Metodologia) serão detalhadas todas as questões propostas, bem como todas as soluções desenvolvidas. Na seção IV (Resultados) serão apresentadas todas as imagens geradas pelos algoritmos, além da discussão sobre os resultados apresentados. E finalizando, na seção V (Conclusão) será dado um parecer sobre o experimento realizado e as considerações finais.

III METODOLOGIA

III.A Questão 1.1

Na questão é solicitada a criação de uma função `dec_int` que receba como parâmetros uma imagem e um valor N múltiplo de 2. A função deve então reduzir a imagem por esse fator e logo após interpolá-la para o tamanho original, preenchendo os espaços com o pixel mais próximo.

Dessa forma, para resolver esse problema, foi criada uma matriz para a imagem reduzida chamada `resized`, que possuirá dimensões $A/N \times L/N$, com A e L sendo respectivamente a altura e largura iniciais da imagem dada. Com isso, basta

percorrer a matriz da imagem original utilizando dois loop for, pulando de N em N elementos e copiando os pixels para a matriz resized. Após isso, é preciso interpolar a imagem para o tamanho original. Cria-se então a matriz que receberá a imagem interpolada, chamada *interpolated*, com dimensões A x L (mesma dimensão da imagem passada como parâmetro), que também através de dois loops for será preenchida. Para tal, basta copiar um pixel da matriz resized N vezes em todas as direções para a matriz *interpolated*, resultando em N^N pixels iguais. A função então retorna dois argumentos, a matriz resized e a matriz *interpolated*.

III.B Questão 1.2

Nessa questão é pedida a criação de uma função *egde_improv* que realize um filtro de aguçamento no domínio espacial que de certa forma melhore a qualidade de uma imagem em qualquer aspecto.

Portanto, foi utilizada uma função gamma que aplica a determinada transformação para cada pixel:

$$T = \text{Max} * (P/\text{Max})^G$$

Em que T é o pixel resultante, Max é o valor de brilho máximo presente na imagem, P o pixel de entrada e G o valor de gamma. Dessa forma, foi definido o valor padrão de G para 2, especificamente para essa questão, de forma arbitrária, para observar o efeito produzido sobre a imagem. Para armazenar esses pixels transformados, é criada a matriz *improved*, que ao final da função será o argumento de retorno.

III.C Questão 1.3

Após a criação das funções anteriores, é pedido então a criação de um programa principal denominado *Prog1*, que passe a imagem de teste fornecida para a função *dec_int* e mostre o resultado obtido. Em seguida, deve-se passar a mesma imagem original pela redução e interpolação bicúbica. Com o resultado dessas duas operações, ou seja, duas imagens interpoladas, deve-se utilizar a função *egde_improv*.

Inicialmente, definiu-se um valor arbitrário múltiplo de 2 (N = 6) para ser o nosso fator de redução. Assim, chama-se a função *dec_int* passando a imagem lida e o argumento n. Logo após, utilizando a função *resize* da biblioteca OpenCV, é realizada a redução e interpolação bicúbica. Posteriormente, utilizando a função *egde_improv*, é realizado o filtro descrito em ambas as imagens resultantes, e mostrado todas as imagens geradas durante o processo.

III.D Questão 1.4

Por fim, é pedida uma alternativa para a função *dec_int*, que de forma esperada, não é a melhor alternativa se o objetivo é tornar a imagem interpolada mais próxima a original. Dessa forma, um meio de fazer essa operação ter melhores resultados seria ao invés de copiar o pixel mais próximo, realizar a média entre o pixel anterior e o próximo, ocorrendo uma passagem mais suave de uma cor para outra, o que no fim provavelmente geraria bordas menos distinguíveis, porém formas retangulares menos aparentes.

III.E Questão 2.1

Nessa questão é pedido para realizar a transformação gamma nas 3 imagens em escala de cinza solicitadas (*car*, *crowd* e *university*) com dois valores maiores que um e dois valores menores que um, selecionando a que obtiver melhor resultado.

Para isso, foi reaproveitado o código da questão 1.2 com uma pequena alteração: a de não usar somente um, mas sim quatro valores distintos para a variável G, de forma a poder selecionar o melhor resultado. Criou-se então uma função denominada *power-law*, que mostra na tela o resultado da transformação da imagem para os valores de gamma 0.3, 0.8, 1.5 e 2.0.

III.F Questão 2.2

No enunciado é requisitado que desta vez, seja realizado o processo de equalização de histograma para as três imagens citadas, devendo-se escolher uma para mostrar o histograma e a CDF antes e depois da transformação.

Para a equalização, primeiramente converte-se a imagem passada para a escala de cinza utilizando a função *cvColor*, e logo após utiliza-se a função *equalizeHist* para gerar o histograma equalizado. Foi então encapsulado essas duas operações na função *equalize*, que retorna a imagem já equalizada.

Prosseguindo com o pedido, é preciso mostrar os histogramas e CDFs. Para tal, foi utilizada da biblioteca Matplotlib o *pyplot*, um conjunto de funções que auxilia na plotagem de gráficos. Com isso em mente, calculamos o histograma, tanto da imagem original quanto da equalizada com o auxílio da função *calcHist*, retornando matrizes que em seguidas foram plotadas no programa. Para calcular as CDFs foi utilizada a função da biblioteca Numpy *cumsum()*, que ao ser aplicada nos histogramas, nos retorna a CDF de ambos. Novamente são plotados as matrizes geradas.

IV RESULTADOS

Após a descrição da metodologia desenvolvida em cada questão, a seguir serão apresentadas as imagens geradas por cada algoritmo a partir de suas entradas.

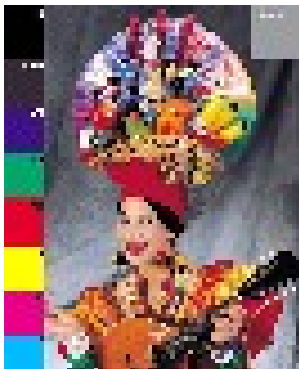
IV.A Questão 1

Foi passada então a seguinte imagem para o *Prog1*:



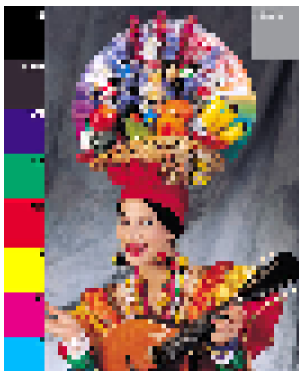
1 Imagem de entrada

Chama-se então a função `dec_int`, que inicialmente realizando a redução, gera a seguinte imagem (ampliada no relatório para fins de visualização):



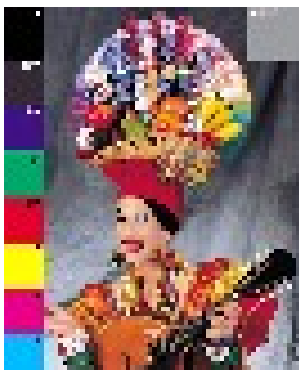
2 *Imagem reduzida para $N = 6$*

A seguir a imagem anterior após a interpolação:



3 *Imagem interpolada*

Após isso, realiza-se a redução cúbica na imagem de entrada inicial (Ampliada no relatório para fins de visualização):



4 *Imagem reduzida de forma cúbica*

Em seguida, interpolou-se de forma cúbica a imagem reduzida acima:



5 *Imagem Interpolada de forma cúbica*

Por fim, as duas imagens interpoladas após o filtro de aguçamento:



6 *Imagem gerada pelo `dec_int` após o filtro `egde_improv`*



7 *Imagem interpolada de forma cúbica após o filtro `egde_improv`*

IV.B Questão 2

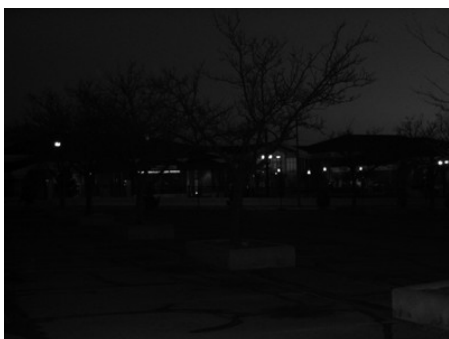
Foram passadas as seguintes imagens para o Prog2:



8 *Imagem de entrada "car"*



9 Imagem de entrada “crowd”



10 Imagem de entrada “university”

Foi aplicada a transformação gamma nos quatro valores descritos e então seleccionados os melhores resultados para cada imagem:



11 Imagem “car” para Gamma = 2.0



12 Imagem “crowd” para Gamma = 0.3



13 Imagem “university” para Gamma = 0.3

Passando para a segunda parte da questão, todas as imagens iniciais foram equalizadas:



14 Imagem “car” equalizada

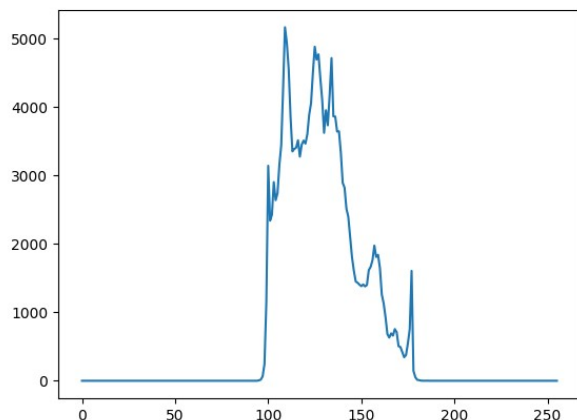


15 Imagem “crowd” equalizada

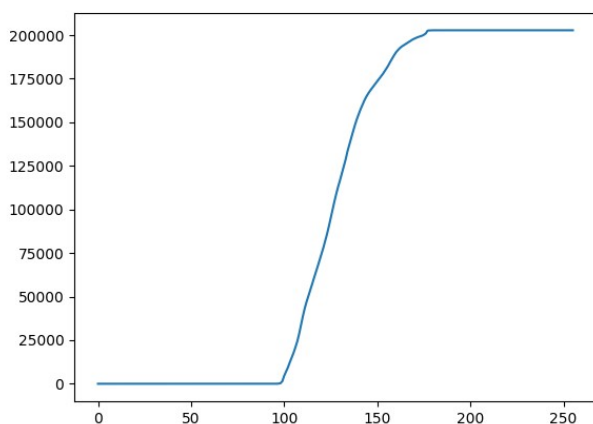


16 Imagem “university” equalizada

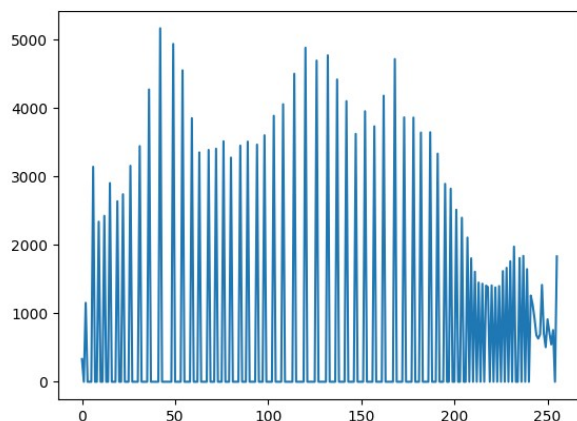
Finalmente, foi selecionada uma das imagens para plotar seus histogramas e CDFs antes e depois da equalização. A imagem escolhida foi a “car”:



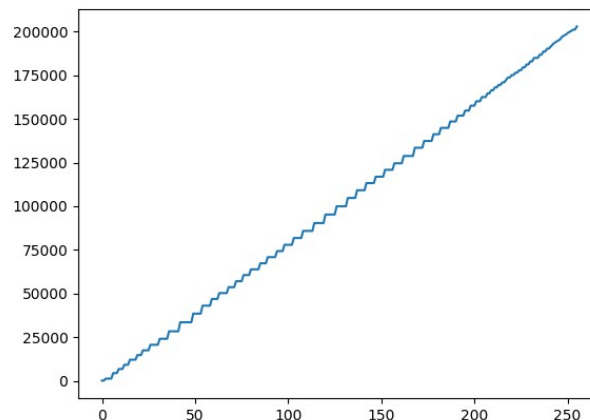
17 Histograma da imagem “car” original



18 CDF da imagem “car” original



19 Histograma da imagem “car” equalizada



20 CDF da imagem “car” equalizada

V CONCLUSÃO

Vê-se através desses experimentos, portanto, que a forma como uma transformação é realizada impacta significativamente no resultado.

Observando o experimento 1, é perceptível que a interpolação cúbica trouxe uma fidelidade maior com a imagem original, já que aproveitou melhor os dados que tinha a disposição.

Com relação ao experimento 2, fica claro que a transformação por equalização de histograma mostrou-se mais eficiente, tendo em vista que possibilitou uma maior visualização dos detalhes de cada imagem. Analisando também os gráficos de histograma, ocorre o comportamento esperado de maior distribuição dos pixels pelo nível de brilho. Analisando a CDF original, nota-se que a imagem aparenta ter sofrido uma transformação linear por partes (contrast stretching), porém após a equalização, o gráfico torna-se semelhante a uma reta, o que comprova novamente uma distribuição bem equalizada dos pixels da imagem.

Enfim, conclui-se que as transformações no domínio do espaço são de suma importância para a visualização de imagens pelo olho humano, que possuindo certas limitações, não consegue observar diversos detalhes, que mesmo presentes nas imagens, aparentavam ser de difícil visualização.

REFERÊNCIAS

- 1 Slides do professor Bruno Macchiavello.
- 2 Site de documentação do OpenCV
- 3 Site de documentação do Matplotlib