

Documentação Trabalho 1 - IPI

Bruna Azambuja
Departamento de Ciência da
Computação
UnB
Brasília, Brasil
bubu.azambuja@hotmail.com

Keywords - processamento, imagens.

I. RESUMO

Este trabalho teve como objetivo principal aplicar os conhecimentos obtidos em sala de aula sobre processamento e características gerais de uma imagem; tais como a distribuição do brilho, cor, contraste e redimensionamento de imagens. Os códigos foram desenvolvidos em MatLab 2018, e testados com imagens distintas, como será explicado melhor na seção de Resultados.

O exercício consiste em duas partes: A primeira testa a capacidade de redistribuir o brilho em níveis, e alterar o tamanho da imagem para uma versão maior (ou menor) dela mesma, adicionando ou retirando pixels. Já a segunda parte do trabalho tem objetivo de testar um algoritmo que tem como princípio testes de brilho e distância para ir de um ponto a outro dentro da imagem – fixando os pontos de chegada e de saída.

As principais dificuldades enfrentadas durante o desenvolvimento foi a utilização da linguagem e implementação de funções já existentes, como será exemplificado mais a frente.

II. INTRODUÇÃO

A. Primeiro Exercício: Redistribuição de Brilho e Redimensionamento de Imagens

Neste exercício foi dada uma imagem qualquer, podendo ser tanto sua versão colorida (RGB) como sua versão monocromática (níveis de cinza); bem como um número inteiro entre 1 a 8 que chamaremos de Nbrilho e um número de ponto flutuante que chamaremos de Fator, onde o primeiro representa a quantidade de níveis de brilho que a imagem de saída deve ter e o segundo é o fator de redimensionamento da imagem.

A saída gerada pelo programa deve ser uma imagem que possui $[2^{Nbrilho}]$ níveis de brilho redistribuídos entre si, de tamanho $[Fator \times Imagem\ original]$.

Note que o fator é a porcentagem de (de)crecimento, ou seja, 1.75 significa um acréscimo de 75% na imagem. Portanto fatores menores ou maiores que um representam imagens menores ou maiores que a original, respectivamente.

B. Segundo Exercício: Algoritmo do Melhor Caminho

Já na segunda parte do trabalho, tem-se um algoritmo que auxilia um time de astronautas em sua missão, dada uma imagem RGB do mapa de Marte, em que um astronauta deve

ir de um ponto a outro, sabendo que os percursos mais claros (de maior brilho) são proporcionalmente mais difíceis de serem percorridos.

Assim, o algoritmo deve traçar o caminho que deveria ser feito pelo astronauta de modo que seja o melhor possível, ou seja, de menor gasto de combustível. Para isso, o algoritmo leva em conta a distância de cada pixel vizinho ao atual até o pixel de chegada, e seu brilho, que representa a dificuldade do caminho.

Após a implementação do programa, conseguimos analisar se é realmente possível chegar ao ponto onde queremos seguindo este algoritmo, e se sim, qual o caminho dado.

III. METODOLOGIA

Dado o resumo do objetivo de cada exercício, será feita a explicação sucinta de como foi alcançado tal objetivo. Quais funções foram usadas e quais algoritmos foram implementados para que o programa funcionasse do jeito esperado.

Lembrando aqui que os nomes de funções utilizadas são referentes ao MatLab 2018.

Primeiro Exercício:

A. Níveis de Brilho

Para a redistribuição dos níveis de brilho foram feitas as seguintes operações:

Primeiro o brilho de cada pixel da imagem foi dividido pelo número de bits que a imagem de saída deveria ter, pixel por pixel. O número gerado foi arredondado pois o brilho não pode ser ponto flutuante, e depois multiplicado pelo mesmo número. Dessa forma, temos uma imagem de 8 bits, porém com o brilho redistribuído pela quantidade de níveis determinada, com seus valores de brilho pixel por pixel truncado para o nível do qual pertence.

Exemplo: se Nbrilho for igual a 1, temos que ao invés de 256 níveis (2^8), teremos apenas 2 níveis (2^1), onde os outros 7 bits não foram necessariamente apagados, apenas redistribuídos em dois níveis, assumindo novos valores dentro de cada nível – nesse caso, 0 para qualquer pixel com valor entre 0 e 127, e 255 para qualquer pixel com valor entre 128 e 255.

Tal funcionalidade poderia ser simplesmente substituída pela função *imapprox* do MatLab, que recebe a imagem original e a quantidade de bits que a saída deve possuir, e retorna a imagem de saída com os níveis de brilho truncados, na quantidade de níveis desejada.

B. Redimensionamento

Para o redimensionamento da imagem foram realizadas as seguintes operações:

Primeiro, criamos uma matriz de zeros com o tamanho do fator de redimensionamento, ou seja, uma matriz vazia [fator x Tamanho Original].

Para preenchê-la utilizamos um algoritmo que percorre a matriz zerada pixel por pixel, e a preenche com o valor do pixel da imagem original mais próximo a ele – onde a linha/columna do pixel é acessada por linha/columna da saída dividida pelo fator de redimensionamento, e arredondado no final com a função *round*. Ou seja, estaremos copiando linhas e colunas e as reproduzindo na quantidade desejada dentro da imagem de saída.

Caso o fator seja menor que um, estaremos na prática, jogando uma determinada quantidade de informação fora, para que a saída seja menor que a imagem original. Já no caso em que o fator é maior que um, na prática estaremos reproduzindo cada pixel da imagem original mais de uma vez, dessa forma, gerará uma saída maior que a imagem original.

Tal funcionalidade poderia simplesmente ser substituída pela função *resize* do MatLab, que recebe a imagem original e o fator de redimensionamento e retorna uma imagem de saída proporcionalmente menor/menor que a entrada.

Segundo Exercício:

C. RGB para Níveis de Cinza

Antes de aplicar o algoritmo de Melhor Caminho, a imagem recebida foi transformada em sua versão monocromática, ou seja, o mapa de Marte RGB recebido, foi processado para que a saída seja uma versão a níveis de cinza desta.

Para isso, utilizamos três matrizes do tamanho da imagem, onde cada uma contém apenas o valor em sua respectiva cor. Ou seja, uma matriz com os valores dos pixels na cor vermelha, uma na azul e uma na verde. Após criadas essas matrizes, criamos outra matriz zerada do mesmo tamanho da original.

A matriz de saída foi preenchida pixel por pixel somando-se os valores das três matrizes de cores, cada uma multiplicada por uma constante determinada.

$$Saída = a.vermelho + b.azul + c.verde \quad (1)$$

Onde os valores de *a*, *b* e *c* foram obtidos no site de documentação da função *RGB2Gray*, e valem respectivamente 0.2989, 0.1140 e 0.5870.

Tal funcionalidade poderia simplesmente substituída pela função *RGB2Gray* do MatLab, que recebe a imagem original e retorna sua versão monocromática.

D. Algoritmo de Melhor Caminho

Após transformada para níveis de cinza, utilizou-se a função *histeq* do MatLab para gerar a imagem equalizada. De acordo com o exercício, tem-se que o brilho da imagem equalizada é proporcional ao nível de dificuldade de percorrer o terreno, logo vamos adotar a seguinte lógica:

Dado o pixel de sua localização atual, calculamos a distância euclidiana para cada um dos 8 pixels vizinhos, e armazenamos em uma matriz [8x4], que contém as

coordenadas *x* e *y*, seu valor de brilho e sua distância ao ponto de chegada.

Com a matriz de distâncias dos vizinhos pronta, foi criada outra matriz [3x4] que contém os 3 menores vizinhos com respeito à distância. Dado esses 3 vizinhos restantes, selecionamos o de menor brilho; esse pixel será nosso ‘próximo passo’, ou seja, o caminho passará por ele e o ‘pixel atual’ passa a ser o selecionado. O pixel selecionado tem seu valor de brilho alterado, para que se possa enxergar o caminho percorrido na imagem de saída.

O algoritmo é repetido até que cheguemos no ponto de chegada.

IV. RESULTADOS

Após uma noção geral de cada função dos exercícios e como foi implementada cada uma, será dada nessa seção a saída gerada de cada um dos programas com imagens pré-definidas.

A. Primeiro Exercício:

A imagem de entrada em nível de teste foi:

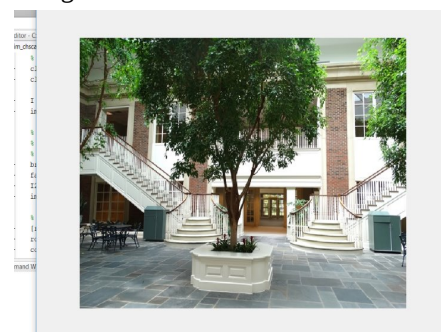


Fig.1. Imagem de entrada

Exemplo 1:

Número de níveis utilizado foi 3, e o fator de redimensionamento foi 1.75.

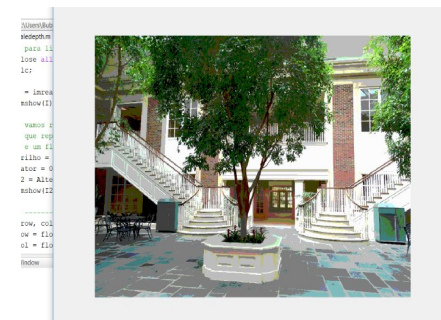


Fig.2. Imagem de Saída1 de Níveis de Brilho 3

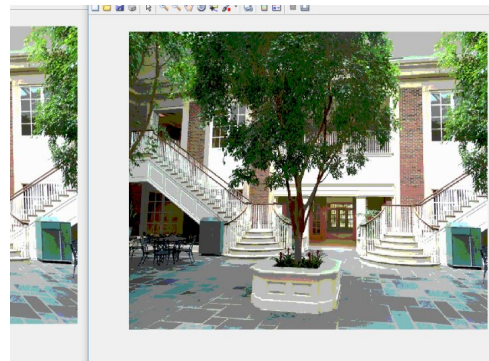


Fig.3. Imagem de Saída2 de Redimensionamento 1.75

Observação: O fator de redimensionamento aponta para um acréscimo de 75% na imagem original, porém a escala da imagem no relatório não condiz com a escala real gerada pelo programa!

Exemplo 2:

Número de níveis utilizado foi 5, e o fator de redimensionamento foi 0.5.

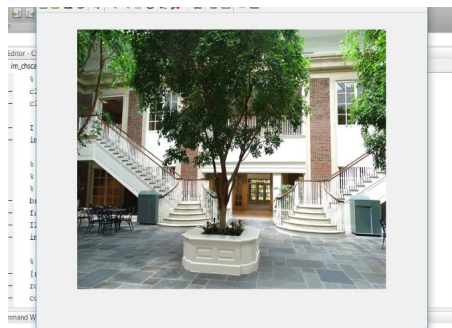


Fig.4. Imagem de Saída1 de Níveis de Brilho 5

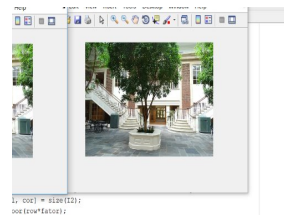


Fig.5. Imagem de Saída2 de Redimensionamento 0.5

Observação: O fator de redimensionamento aponta para um decréscimo de 50% na imagem original, porém a escala da imagem no relatório não condiz com a escala real gerada pelo programa!

B. Segundo Exercício

A imagem de entrada em nível de teste foi:

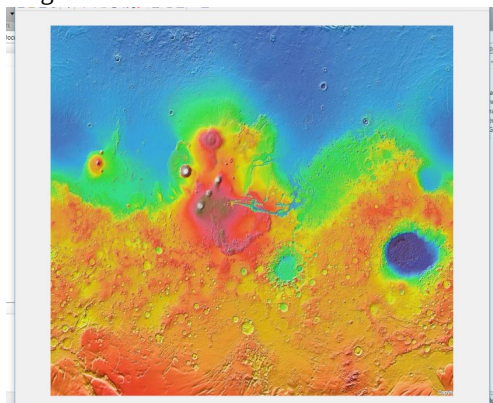


Fig.6. Imagem de Entrada

Imagem do mapa de Marte formato RGB.

Após a transformação de RGB para níveis de cinza:

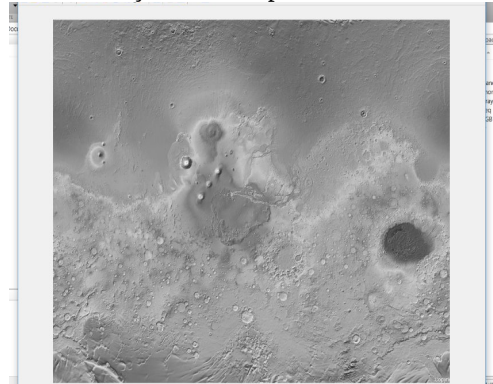


Fig.7. Versão Monocromática da imagem de entrada

Após a equalização do histograma utilizando a função *histeq*:

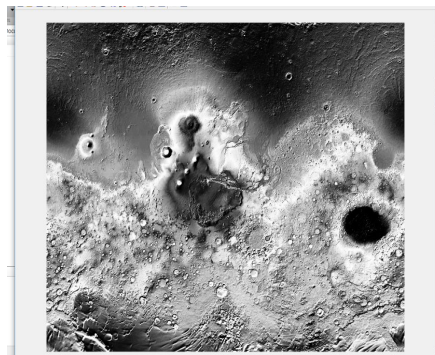


Fig.8. Histograma Equalizado

Saída do Algoritmo de Melhor Caminho:

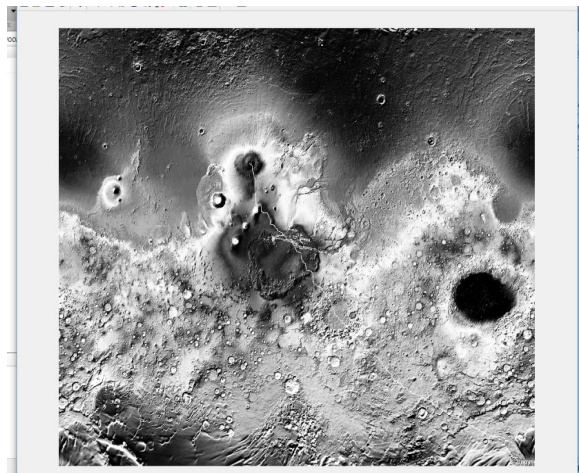


Fig.9. Melhor Caminho Traçado

Nesta imagem acima conseguimos ver o melhor caminho pintado em branco para chegar do ponto de saída até o ponto de chegada.

CONCLUSÃO

Do primeiro exercício, pôde-se perceber que a imagem perde alguns detalhes com o redimensionamento, causando um agravamento das bordas, mas que, num quadro geral, não há tanta perda de informação em nenhum dos dois casos – para imagens maiores e menores que a original.

E do tratamento dos níveis de brilho, pôde-se ver que para perceber de fato uma diferença entre a imagem original e a gerada pelo programa, a quantidade de níveis deve ser bem menor que à padrão. Ou seja, para números mais altos (como 2^5 níveis) quase não foi possível notar a diferença, porém para números mais baixos (como 2^3 níveis) já nota-se irregularidades na imagem que não vemos na original.

Do segundo exercício pôde-se concluir que, de fato, seguindo o algoritmo explicitado, é possível os astronautas chegarem de um ponto ao outro seguindo um caminho ótimo com respeito a dificuldade do percurso e distância a percorrer.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Slides do professor Bruno Macchiavello.
- [2] Site de documentação do MatLab – MathWorks.