

# Projeto 1

Gustavo Antonio Souza de Barros  
18/0064487

Introdução ao processamento de imagens  
Engenharia da Computação  
Universidade de Brasília  
Brasília, Distrito Federal 70910-900  
Email: gustavoasb@gmail.com

**Abstract**—O projeto desenvolvido tem como objetivo a realização de duas atividades. A primeira tem foco na manipulação de imagens, criando uma função que mude a quantidade de bits da imagem, assim como seu tamanho. A segunda aborda sobre um problema dado, em que um conjunto de astronautas tem de obter uma amostra de solo de um local específico de Marte. Através de uma imagem aérea RGB, o objetivo é achar o jeito mais rápido de chegar a esse local usando a menor energia possível.

## I. INTRODUCTION

O processamento de imagens é um tópico que está sempre em pauta quando se trata de novas tecnologias. O crescimento dessa área é exponencial e compreende diversas aplicações.

A capacidade de manejar imagens a fim de adquirir dados ou implementar funcionalidades é extremamente importante, tornando o projeto essencial para o treinamento desta capacitação.

A fim de entender melhor a finalidade da primeira tarefa, é necessário ter conhecimento sobre o que são os bits quando se fala de imagem. Um bit é o que chamamos de profundidade de cor, de um jeito simplificado, descreve a quantidade de cores que os pixels da imagem podem ser representados. Através da equação abaixo é fácil perceber como isso se aplica.

$$Cores = 2^n \quad (1)$$

Na equação 1,  $n$  representa o número de bits, logo podemos montar o esquema abaixo:

- 1 bit,  $2^1 = 2$  cores, imagem binária.
- 2 bit,  $2^2 = 4$  cores.
- 3 bit,  $2^3 = 8$  cores.
- 4 bit,  $2^4 = 16$  cores.
- 5 bit,  $2^5 = 32$  cores.
- 6 bit,  $2^6 = 64$  cores.
- 7 bit,  $2^7 = 128$  cores.
- 8 bit,  $2^8 = 256$  cores.

Em imagens não coloridas, nós assumimos que os pixels tem o que chamamos de intensidade, que se refere ao brilho do mesmo. Um pixel com brilho 255 em uma imagem com 256 bits é branco, e um com 0 é preto. Entre esses vários pixels estão os tons de cinza, que são determinados pela quantidade de bits da imagem.

Em imagens coloridas, nós entramos no tópico do RGB (Red, Green, Blue), que são como as cores de uma imagem são

apresentadas. Cada cor tem uma quantidade de cada um desses componentes e assim podemos representa-las com valores destes. Para esse tipo de imagem, a quantidade de bits se refere a quantidade máxima que cada componente pode chegar.

## II. METODOLOGIA

### A. Questão 1

Para a primeira atividade, foram desenvolvidas rotinas que alteram a quantidade de bits de uma imagem, assim como suas dimensões. O código fonte está salvo como `question1.py` e pode ser executado através da linha de comando `python question1.py`.

Primeiramente, lê-se a imagem teste, com auxílio do OpenCV. Após isso, foi implementada uma função que muda a quantidade de bits da imagem através da entrada do usuário, através de uma fórmula matemática que faz isso automaticamente.

$$\frac{pixelvalue}{256/2^n} \quad (2)$$

Sendo  $n$  a quantidade de bits desejada.

Após a conversão realiza-se arredondamentos e adapta-se a imagem de volta para a quantidade de bits padrão, 256. Esse procedimento é adotado pois o foco é apenas demonstrar a diferença entre as imagens de bits diferentes, e não convertê-las em si.

A segunda parte dessa atividade é baseado no redimensionamento de uma imagem, o usuário dará um valor que definirá as novas dimensões da mesma.

$$H_{nova} = H_{original} \cdot k \quad (3)$$

$$L_{nova} = L_{original} \cdot k \quad (4)$$

Sendo  $k$  o valor que definido para mudar o tamanho da imagem. Por exemplo,  $k = 0.5$  retornaria uma imagem com metade do tamanho original.

Para  $k$  menor que 1, pode-se simplesmente repetir os pixels da original de forma a jogar alguns valores fora. Porém para valores elevados de  $k$  nós deparamos com o problema de espaços vazios. Para resolver isso repete-se a última linha lida e coloca-se no espaço vazio encontrado. O mesmo procedimento é efetuado para colunas, fechando assim a imagem redimensionada.

### B. Questão 2

Para a segunda atividade, lê-se o mapa aéreo de marte com cores RGB com intensidades a indicar a energia gastada para passar naquele local. Tira-se dali um caminho curto e com gasto de energia mais baixo possível.

O primeiro procedimento adotado é converter a imagem de RGB para monocromática. Isso é facilmente resolvido usando a fórmula:

$$Intensidade = \frac{R + G + B}{3} \quad (5)$$

Realizando a rotina para todos os pixels consegue-se uma imagem em tons de cinza.

Após isso, com ajuda do OpenCV realiza-se a equalização do histograma da imagem, a fim de enaltecer os detalhes, através do aumento do contraste.

Com origem e destino definidos, inicia-se um loop que altera o pixel em que o algoritmo está atualmente localizado, e que será finalizado quando é alcançado o pixel de destino.

O laço anteriormente citado funciona da seguinte maneira, os 8 pixels ao redor do atualmente são os candidatos a serem escolhidos como próximo do caminho. Mas para isso existem condições, primeiro efetua-se a distância euclidiana entre os 8 e o pixel de destino, através da equação:

$$Distancia = \sqrt{(x - x_f)^2 + (y - y_f)^2} \quad (6)$$

Depois dos cálculos, efetua-se uma ordenação pela distância e as três menores vão para a próxima etapa.

Com os pixels de menor distância ajustados, faz-se um teste com estes e é selecionado o de menor intensidade. Esse selecionado é o novo pixel que irá compor o caminho até a localização final, e a partir dele a fase de seleção iniciará de novo, processo que se repete inúmeras vezes, de forma a alcançar o local almejado.

### III. RESULTADOS

Essa seção abordará sobre os resultados encontrados nos tópicos debatidos anteriormente.

#### A. Questão 1

A imagem original dos testes é mostrada abaixo para efeito comparativo.



Fig. 1. Imagem original

O primeiro teste da atividade inicial é com um valor de redimensionamento menor que 1. Para ele as entradas foram (5,0.5):



Fig. 2. Saída 1

A diferença entre a original e a do primeira teste é mínima, pois trata-se de diminuição a diferença de bits não é tão chamativa.

O segundo teste foi efetuado usando as entradas (3,1.75):



Fig. 3. Saída 2

No teste 2 já é possível notar a diferença nas cores, que perdem detalhe, e se aberta a imagem original seria mais clara a perda de qualidade da imagem ao ser aumentada.

Para um última checagem, testa-se um imagem em tons de cinza. As entradas foram (2,1)



Fig. 4. Saída 3

é evidente a transformação das cores ao usarmos apenas 2 bits. A imagem não foi redimensionada pelo coeficiente ser 1.

#### B. Questão 2

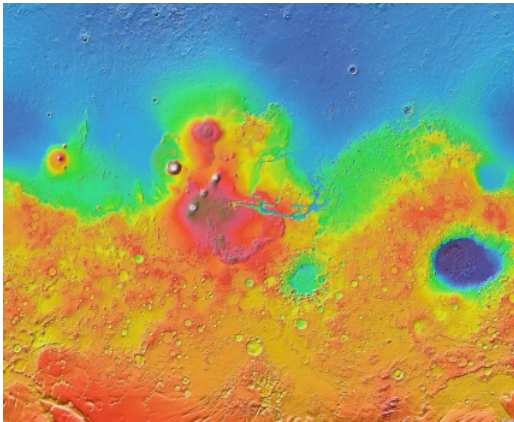


Fig. 5. Mapa aéreo de Marte, Questão 2

Através do mapa aéreo acima, acha-se o caminho usando o algoritmo explicado anteriormente, tendo como saída:

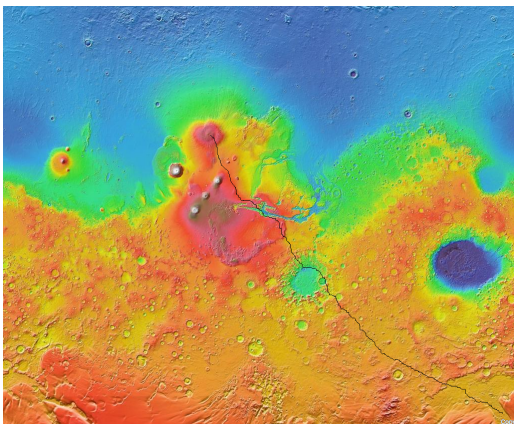


Fig. 6. Saída, Questão 2

O caminho marcado em preto foi o encontrado como o balanço entre distância e gasto de energia.

#### IV. CONCLUSÕES

Levando em conta toda a realização do projeto, considera-se que este foi executado com sucesso e com resultados aceitáveis. Apesar de ter sido concluído de forma decente, há espaço para melhores, existem algoritmos de redimensionamento extremamente superiores, além de execuções que poderiam ser muito mais rápidas se fossem trabalhadas de forma mais minuciosa.

Os resultados deram a perceber na prática como funcionam diversos dos conceitos desenvolvidos em sala de aula, além de uma maior percepção do assunto.

Assuntos como o aumento ou diminuição de imagens de forma a deixar a imagem de forma mais agradável possível fora um assunto interessante, pois faz-se constatar como funcionam os melhores algoritmos para essas rotinas.

#### REFERENCES

- [1] OpenCV Team, *Open Source Computer Vision Documentation*, opencv.org.
- [2] Wikipedia, *Color depth*. Accessed in 20, April, 2019.
- [3] SciPy Team, *NumPy User Guide*. docs.scipy.org.
- [4] Paul Barrett, *Euclidean Distance*, September, 2015, www.pbarrett.net/techpapers/euclid.pdf.
- [5] Gerald Bakker, *Color Coding and the RGB Color Space*, December, 2014, geraldbakker.nl/psnumbers/rgb-explained.html.