Universidade de Brasília
Departamento de Ciência da Computação (CIC)
Algoritmos e Programação de Computadores
Prof. Dr. Vinícius Ruela Pereira Borges

Projeto II

Data de divulgação: 09/06/2018

Objetivo

Elabore um programa que leia uma imagem digital colorida (sob o formato Red-Blue-Green RGB) em disco e realize um processamento de forma a alterar suas cores, produzindo uma "arte" na imagem como ocorrem em alguns aplicativos em smartphones. A Figura 1 ilustra tal processamento. Observe que a imagem digital original na Figura 1(a) possui várias cores diferentes, principalmente entre as regiões de transição de objetos, como as nuvens e o céu. Um processamento de forma a produzir a "arte" é mostrado na Figura 1(b), em que a imagem está representa em apenas 8 cores.

Figura 1: (a) Imagem digital original (b) Imagem digital processada.

Sua tarefa é implementar o algoritmo K-Médias, bastante popular na Ciência da Computação, para processar a imagem conforme a Figura 1.

Uma imagem digital é uma representação discreta de uma imagem em um domínio originalmente contínuo. O processo de discretização pode ser ilustrado conforme a Figura abaixo:

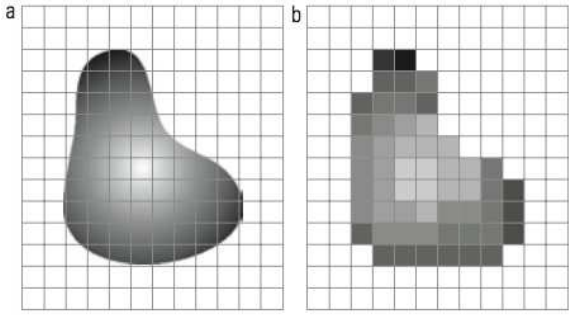

Figura 2: (a) Objeto e a matriz de pixels (b) Imagem digital contendo o objeto na forma discreta.

Observe que na Figura 2(a), uma matriz de pixels é sobreposta ao objeto (em sua representação contínua) e na Figura 2(b) como é a representação discreta desse mesmo objeto em uma imagem digital. Neste sentido, cada pixel (cada célula) assume uma cor (em nível de cinza, da cor preta à cor branca) dependendo da cor do objeto.

Mais detalhes sobre o programa a ser desenvolvido para elaboração do Projeto II são descritos a seguir:

O que deverá ser implementado

1. Ao iniciar a execução do seu programa, deve-se apresentar um menu de opções ao usuário. O usuário deve digitar uma das opções e após o encerramento da tarefa requisitada, o menu deverá ser novamente exibido ao usuário, exceto se a opção desejada for o encerramento do programa.

   Cada opção do menu corresponde a uma funcionalidade específica do programa de viabilizar o processamento da imagem digital. O menu de opções deve possuir, ao menos, as seguintes opções:

   1. Abrir uma imagem em disco

   O usuário deverá inserir o nome de uma imagem em disco, juntamente com o formato. O **único** formato permitido é o *Portable Pixel Map* (PPM).

   Uma imagem PPM é um arquivo de texto que possui a seguinte organização, mostrada de acordo com o exemplo abaixo:

```
P3
4 4
255
 0   0   0     0   0   0     0   0   0    255   0 255
 0   0   0     0 255   7     0   0   0      0   0   0
 0   0   0     0   0   0     0 255   7      0   0   0
255   0 255     0   0   0     0   0   0      0   0   0
```

   - Primeira linha: apresenta o identificador associado com a codificação do arquivo, podendo ser `P3` ou `P6`. Neste projeto, utilizaremos somente o identificador `P3`, que indica que o arquivo texto associado a imagem possui caracteres codificados em ASCII.
   - Segunda linha: dois valores inteiros separados por um espaço em branco representando a resolução da imagem $n \times m$, isto é, a quantidade de pixels nas direções horizontal e vertical, respectivamente, ou simplesmente, a quantidade de colunas e de linhas. Por exemplo, a imagem associada ao arquivo acima possui $m = 4$ colunas e $n = 4$ linhas, respectivamente.
   - Terceira linha: um número inteiro l associado ao valor máximo de intensidade na imagem. Por exemplo, na imagem associada ao arquivo acima, o valor máximo de intensidade é 255.
   - Quarta linha (em diante): a matriz de pixels em código ASCII, totalizando $n \times m$ pixels, cujos valores variam entre zero e l. Cada pixel é representado por três valores numéricos consecutivos, denotando os valores de intensidade para os canais vermelho, verde e azul. Um valor de intensidade próximo a zero significa que a cor não é utilizada, enquanto que o valor máximo l para um canal de cor indica utilização máxima dessa intensidade. Mais informações podem ser obtidas em[1].

   Para carregar uma imagem digital no disco para a memória, isto é, em uma variável do programa, deve-se utilizar o tipo de variável ponteiro para arquivo (`FILE *`), conforme mostra o trecho de código-fonte a seguir:

```
...
    FILE *fp;
    char nome_arquivo[201];

    /*
       Leitura do nome do arquivo
    */

    fp = fopen(nome_arquivo,"r");

    /*
```

[1] `http://netpbm.sourceforge.net/doc/ppm.html`

```
        Verificacoes necessarias para saber se o arquivo
        foi aberto corretamente
    */

    fclose(fp);

...
```

em que, `fopen` é uma função que abre e habilita o acesso ao conteúdo de um arquivo para uma variável do tipo ponteiro para arquivo, e `fclose` encerra o acesso ao conteúdo do arquivo. O nome do arquivo pode conter, no máximo, 200 caracteres.

O conteúdo do arquivo (que representa a imagem) deverá ser lido e armazenado utilizando um registro (`struct`) e declarando-se um tipo de dados com base nesse registro criado, conforme o comando:

```
struct imagem
{
    /* Campos que caracterizam uma imagem */
};

/* Comando para criar um tipo de dados */
typedef ...
```

Nesse registro, deve-se obrigatoriamente incluir os campos de número de linhas, número de colunas, a matriz de pixels (para cada canal de cor) e o valor máximo de intensidade. Considere que as imagens que seu programa deve manipular tenham, no máximo, a resolução de 2000×2000 pixels.

2. Processar a imagem

O processamento a ser realizado na imagem se baseia no algoritmo K-Médias. Tal algoritmo é bastante utilizado na resolução de diversos problemas na área de Ciência da Computação, mais especificamente na sub-área de aprendizado de máquina e visão computacional. O K-Médias é um método de agrupamento de dados que tem como objetivo encontrar K grupos (padrões ou regiões) na imagem. Estes grupos são representados por centróides, que são médias numéricas de todos os pixels pertencentes aos agrupamentos em questão.

O K-Médias funciona como se segue. Primeiramente, deve-se atribuir valores iniciais aos K centróides $c_1, ..., c_K$, que representam os centros dos respectivos K agrupamentos dados por $C_1, ..., C_K$, onde $K \geq 2$. Em seguida, atribui-se a pixel da imagem o rótulo do grupo que possui o centróide associado mais similar considerando as cores vermelho, verde e azul. Por conseguinte, os centróides tem seus valores atualizados com base nos pixels que passaram a pertencer aos respectivos agrupamentos. Assim, o processo se repete enquanto um critério de parada não for satisfeito. O algoritmo descritivo do K-Médias, aplicado em uma imagem, visando encontrar K grupos de cores distintas, é descrito conforme:

Para calcular a distância entre cada pixel da imagem e cada centróide, deve-se utilizar a distância Euclidana (adaptada para este projeto) como se segue:

$$D_E(I[x][y], c[i]) = \sqrt{\sum_{l \in R,G,B} (I[x][y][l] - c[i][l])^2} \quad (1)$$

em que $I[x][y]$ é a notação discreta para um pixel (x, y) da imagem e c_i é um centróide, para $1 \leq i \leq K$. Considere que o algoritmo deve manipular, no máximo, 100 centróides. O programa deverá solicitar ao usuário que forneça o valor K. Você ficará responsável por adotar a estratégia adequada de como inicializar adequadamente os valores dos K centróides e de como ajustar o valor para T, por meio de experimentos.

Algoritmo 1: Algoritmo descritivo K-Médias.

Entrada: Imagem em três canais de cores $I[2000][2000][3]$, número de linhas, número de colunas, centróides $c[100][3]$, o número de classes K e o número de iterações T
Saída: Nenhuma

```
1  início
2      para cada centróide faça
3          inicialize com valores aleatórios, ou solicite ao usuário que informe os valores, ou coloque
           manualmente quaisquer valores;
4      fim
5      t ← 1 repita
6          Calcule a distância entre cada pixel da imagem e cada centróide;
7          Para cada pixel, atribua a classe m (cujos identificadores são entre 1 ≤ m ≤ K) associada
           ao centróide m que possui a menor distância calculada na etapa anterior;
8          Para cada centróide c_m, atualize seus valores com base na média de todos os pixels
           atribuídos à classe m.;
9          t ← t + 1;
10     até t ≤ T;
11 fim
```

3. Salvar a imagem processada em disco

Após a realização do processamento da imagem pelo algoritmo K-Means, a nova imagem obtida deve ser salva em disco, também no formato .ppm. O usuário deverá fornecer o nome da imagem.

0. Encerrar a execução do programa

O programa deve ter sua execução encerrada quando o usuário acessar essa opção.

2. Para cada funcionalidade do programa, deve-se elaborar, ao menos, uma função. O tipo de retorno da função e a quantidade de argumentos que ela recebe deverá ser objeto de decisão do aluno. O menu de opções também deve ser programado dentro de uma função, e chamado dentro da função `main()`.

3. O programa a ser desenvolvido deve constar de todos os testes verificatórios afim de orientar o usuário no fornecimento correto das informações, bem como informar o usuário quando um valor informado estiver incorreto.

Instruções para envio

Enviar o arquivo de código-fonte implementado com o seguinte nome:

<numero de matrícula>_< primeiro nome e último nome >.c

em que <numero do matrícula> deve ser substituído pelo seu número de matrícula e < primeiro nome e último nome > deve ser substituído pelo nome e último sem espaços em branco. Por exemplo, se o número de matrícula é 10/1587778 e o nome do aluno é Palmério Machado Orvalho, o nome do arquivo será 101587778_PalmerioOrvalho.c.

Para enviar o arquivo compactado, entre no Moodle e procure pelo link (tarefa) "Envio do Projeto II". Entre neste link, faça o upload do arquivo e confirme o envio do arquivo.

IMPORTANTE: Data limite para envio do trabalho: 04 de julho de 2018.

Observações importantes

- Caso o projeto seja entregue fora do prazo estipulado (deadline), este receberá uma redução de 20% na nota, por dia de atraso. Após 5 dias de atraso, não será mais aceito o recebimento do trabalho.
- O aluno que plagiar o projeto prático será automaticamente reprovado na disciplina.
- Os programas devem ser feitos **obrigatoriamente** em linguagem C ANSI.
- Os programas que apresentarem erros no momento da compilação receberão **nota zero**.
- Mantenha seu programa apropriadamente **indentado** e **comentado**, caso contrário sofrerá penalização na nota.