

Homework2

姓名：谢雨汐 学号：1600012860

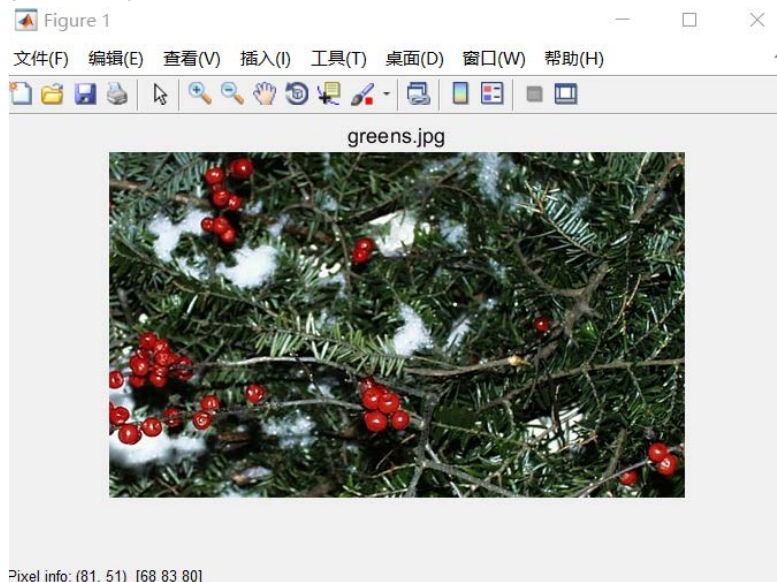
一、题目要求

1. 安装好你的 Matlab 软件，请特别关注图像处理工具包 Image processing toolbox.
2. 用 Matlab 函数把附件的 greens.jpg 图像文件格式转为二进制格式.ppm，并保存为 greens.ppm。
3. 编写你自己的 imread(*.ppm') 函数，来读出步骤 2 生成的.ppm 文件.
4. 提交你的代码，结果和作业报告。

二、问题分析与解答

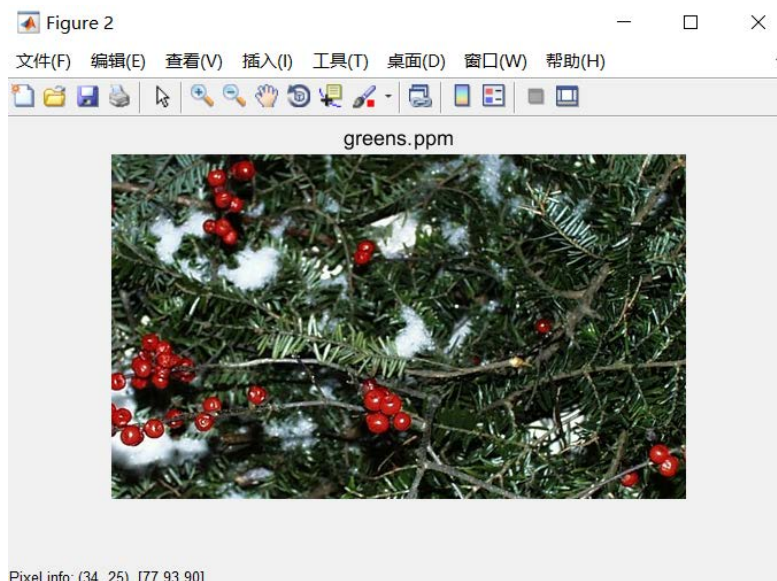
1. 使用 Matlab 函数把 greens.jpg 图像格式转为二进制格式.ppm，并保存为 greens.ppm
[具体代码详见文件 jpg2ppm.m]

1st step. 读取并显示 greens.jpg



2nd step. 将 greens.jpg 图像转为.ppm 格式的 greens.ppm

3rd step. 读取并显示转化后的图像 greens.ppm



2. 编写你自己的 `imread('*.ppm')` 函数，来读出步骤 2 生成的 .ppm 文件

[具体代码详见文件 `my_imread.m`]

1st step. ppm 图像格式分析

pbm: bitmap; pgm: grayscale; ppm: pixmaps;

magic number (first 2 bytes): format (pbm / pgm / ppm) & encoding (ASCII / binary)

Magic Number	Type	Encoding
P1	Bitmap	ASCII
P2	Graymap	ASCII
P3	Pixmap	ASCII
P4	Bitmap	Binary
P5	Graymap	Binary
P6	Pixmap	Binary

ppm $\left\{ \begin{array}{l} \text{head} \left\{ \begin{array}{l} P3 \text{ or } P6 \\ \text{size (ASCII)} \\ \text{maximum pixel value} \end{array} \right. \\ \text{data} \left\{ \begin{array}{l} \text{ASCII: in the order of RGB with space and enter} \\ \text{binary: 24bits/pixel and 8bits/RorGorB} \end{array} \right. \end{array} \right.$

2nd step. 编写 `imread_ppm()` 函数

a) head

提取二进制文件第一行为字符串格式，按空格拆分，即可得到 magic number、size 和 max pixel 四种数据组成的 cell 数组

```
% open the target file in binary mode
fppm = fopen(filename, 'rb');
% get the head of the target file
head = fgetl(fppm);
head = split(head);
ptype = head(1);
ptype = ptype{1, 1};
column = head(2);
column = str2num(column{1, 1});
row = head(3);
row = str2num(row{1, 1});
maxpixel = head(4);
maxpixel = maxpixel{1, 1};
```

b) binary

需要以 8bit 为单位依次读取数据，即可得到每个像素点处的 RGB 值

```
if ptype == 'P6'
    % ppm in binary mode
    for i = 1:row
        for j = 1:column
            for k = 1:3
                node = fread(fppm, 1, 'uint8');
                my_image(i, j, k) = uint8(node);
            end
        end
    end
end
```

c) ASCII

提取二进制文件的数据部分为字符串格式，按空格拆分，即可得到每个像素点处的 RGB 值所组成的数组，根据数值在数组中的编号即可计算出对应像素矩阵中的位置

```
else if ptype == 'P3'
    % ppm in ASCII mode
    % get the data of the target file
    cnt = 1;
    while (~feof(fppm))
        data = strip(fgetl(fppm));
        data = split(data);
        len = length(data);
        for d = 1:len
            k = mod(cnt-1, 3) + 1;
            j = mod((cnt-k)/3, column) + 1;
            i = mod(((cnt-k)/3-j+1)/column, row) + 1;
            node = data(d);
            node = str2num(node{1,1});
            my_image(i, j, k) = uint8(node);
            cnt = cnt + 1;
        end
    end
end
```

3rd step. 测试函数

[具体代码详见文件 **test.m**]

先测试 `imread_ppm()` 函数对 binary mode 的 `greens.ppm` 图像的读取；再将 `greens.ppm` 转换成 ASCII mode 的 `greens1.ppm` 图像，测试 `imread_ppm()` 对 ASCII mode 的图像的读取

