Project 0 Questions

# Instructions

* Compile and read through the included MATLAB tutorial.
* 3 questions.
* Include code.
* Feel free to include images or equations.
* Please make this document anonymous.
* Please use only the space provided and keep the page breaks. Please do not make new pages, nor remove pages. The document is a template to help grading.
* If you really need extra space, please use new pages at the end of the document and refer us to it in your answers.

# Questions

Q2: We wish to set all pixels that have a brightness of 10 or less to 0, to remove sensor noise. However, our code is slow when run on a database with 1000 grayscale images.

*Image:* grizzlypeakg.png

|  |
| --- |
| A = imread('grizzlypeakg.png');  [m1,n1] = size( A );  for i=1:m1  for j=1:n1  if A(i,j) <= 10  A(i,j) = 0;  end  end  end |

1

2

3

4

5

6

7

8

9

Q2.1: How could we speed it up?

B = A<=10;

A(B)=0;

test\_for\_loop历时 9.439035 秒。 logical\_indexing历时 0.032555 秒。

Q2.2: What factor speedup would we receive over 1000 images? Please measure it.

Ignore file loading; assume all images are equal resolution; don’t assume that the time taken for one image ×1000 will equal 1000 image computations, as single short tasks on multitasking computers often take variable time.

A2.2: 将上问操作用分解用循环和逻辑索引进行1000边（循环1000次） 测时间得：

For 循环（10张）：历时 81.672077 秒。

逻辑索引（1000）张：

历时 29.704830 秒。

Q2.3: How might a speeded-up version change for color images? Please measure it.

*Image:* grizzlypeak.jpg

A2.3: Your answer here.

For loop:

一张：历时 8.090302 秒。

10张：历时 81.088474 秒。

Logical indexing :

单独一张：

历时 0.032430 秒。

1000:

历时 30.267958 秒。

Q3: We wish to reduce the brightness of an image but, when trying to visualize the result, all we sees is white with some weird “corruption” of color patches.

*Image:* gigi.jpg

|  |
| --- |
| I = double( imread('gigi.jpg') );  I = I - 20;  imshow( I ); |

1

2

3

Q3.1: What is incorrect with this approach? How can it be fixed while maintaining the same amount of brightness reduction?

A3.1: 当我们用double函数时，仅仅改变了本来是uint8的图像的类型， 数值本身没有变，当我们昨晚reduction后imshow函数以double类型输出图像时将[0,1]之外的数字规整处理，然后我们看到了白色图像，

我们可以不用将图像转换为double，直接在uint8上直接减去20来降低亮度。

Q3.2: Where did the original corruption come from? Which specific values in the original image did it represent?

A3.2: 一张图片有RGB三个颜色通道，若三个通道的像素值都为0，图片中该点呈现为黑色，若全为255（浮点数中1）时白色， 其他情况为彩色，可推断图中的corruption来源于来自图片中brightness reduction 后通过imshow显示图片时 三个通道中颜色不一致的区域 ，例如，图中RGB（39,18,19）的点-20后变为（18，-2，-1）当imshow显示时又便为（1，0，0）呈现为红色的点，很多个这样的点累计形成了显而易见的corruption。