（三）matlab数字图像处理实验-图像灰度变换处理

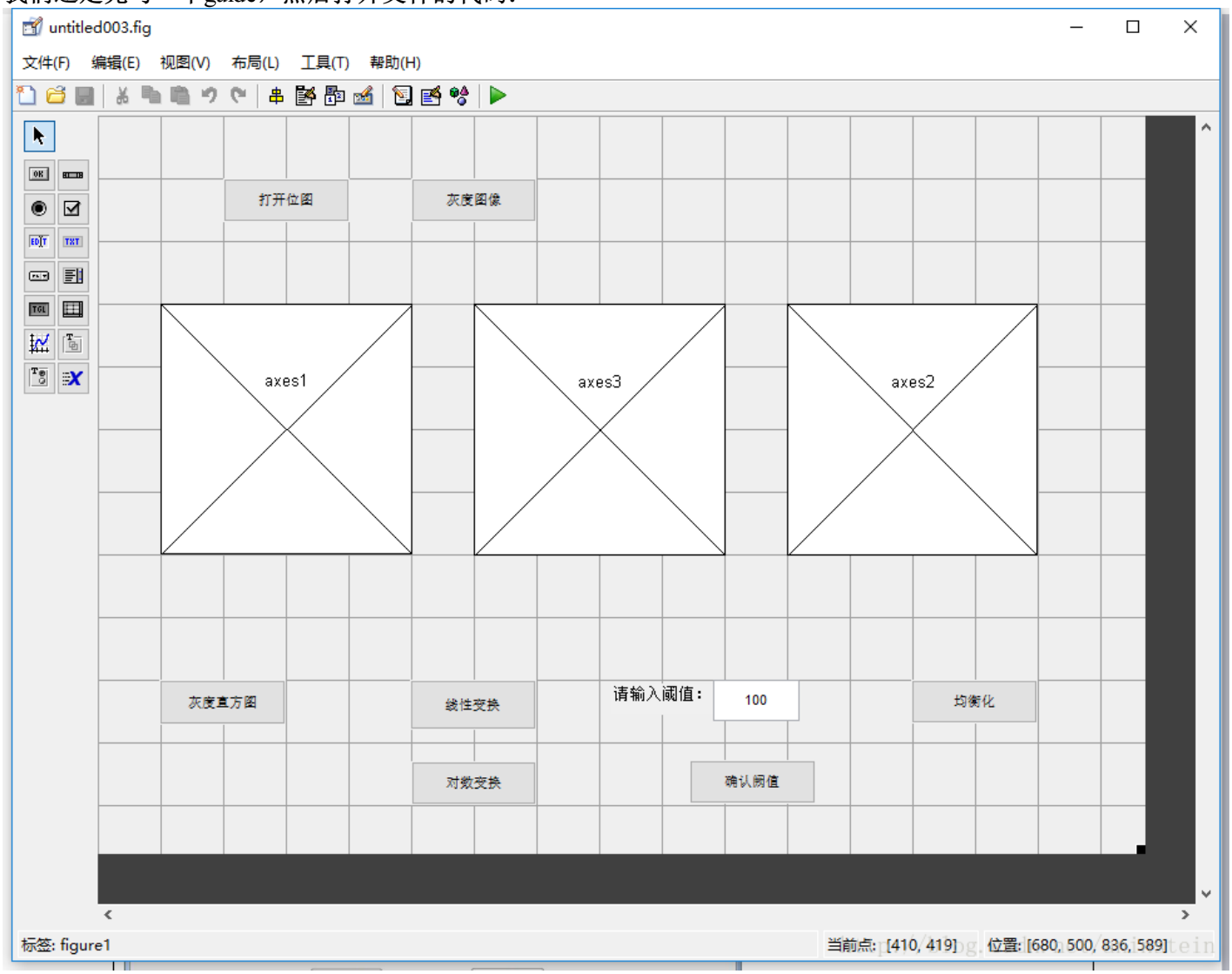
原创

奋斗的昌老师 于 2018-01-06 18:47:37 发布 32438 收藏 142
分类专栏： 数字图像处理 文章标签： matlab 数字图像处理 实验课 灰度变换

前两节都是熟悉一下怎么在matlab底下对图片做一些操作，并没有什么卵用，这一节稍微有点卵用，灰度变换一般是图像处理的第一步。

数字图像处理实验1-9点击下列链接有源码和链接：
matlab数字图像处理实验

我们还是先写一个guide，然后打开文件的代码：



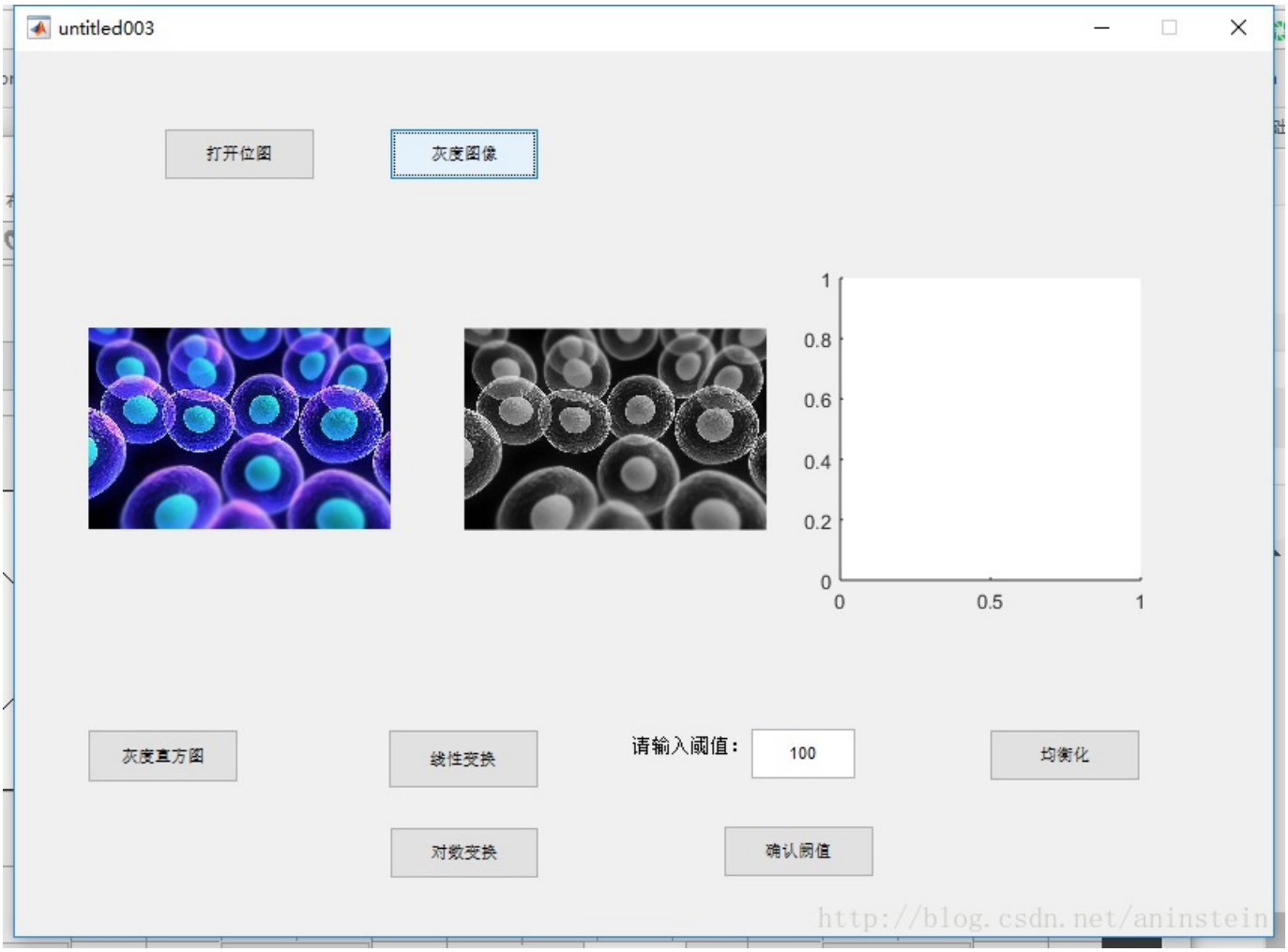
打开文件摁纽的代码：

```
% --- Executes on button press in 打开位图.
function openFile_Callback(hObject, ~, handles)
[filename,pathname]=uigetfile('*.bmp','select image');
str=[pathname filename];
[handles.I,handles.map]=imread(str);
guidata(hObject,handles);
axes(handles.axes1);
imshow(handles.I,handles.map);
```

把图片转换成灰度图片的代码：

```
% --- Executes on button press in 转灰度图.
function togray_Callback(hObject, eventdata, handles)
X=rgb2gray(handles.I);%就是这个函数把图片变成灰度图
guidata(hObject,handles);
axes(handles.axes3);
imshow(X,handles.map);
```

效果：



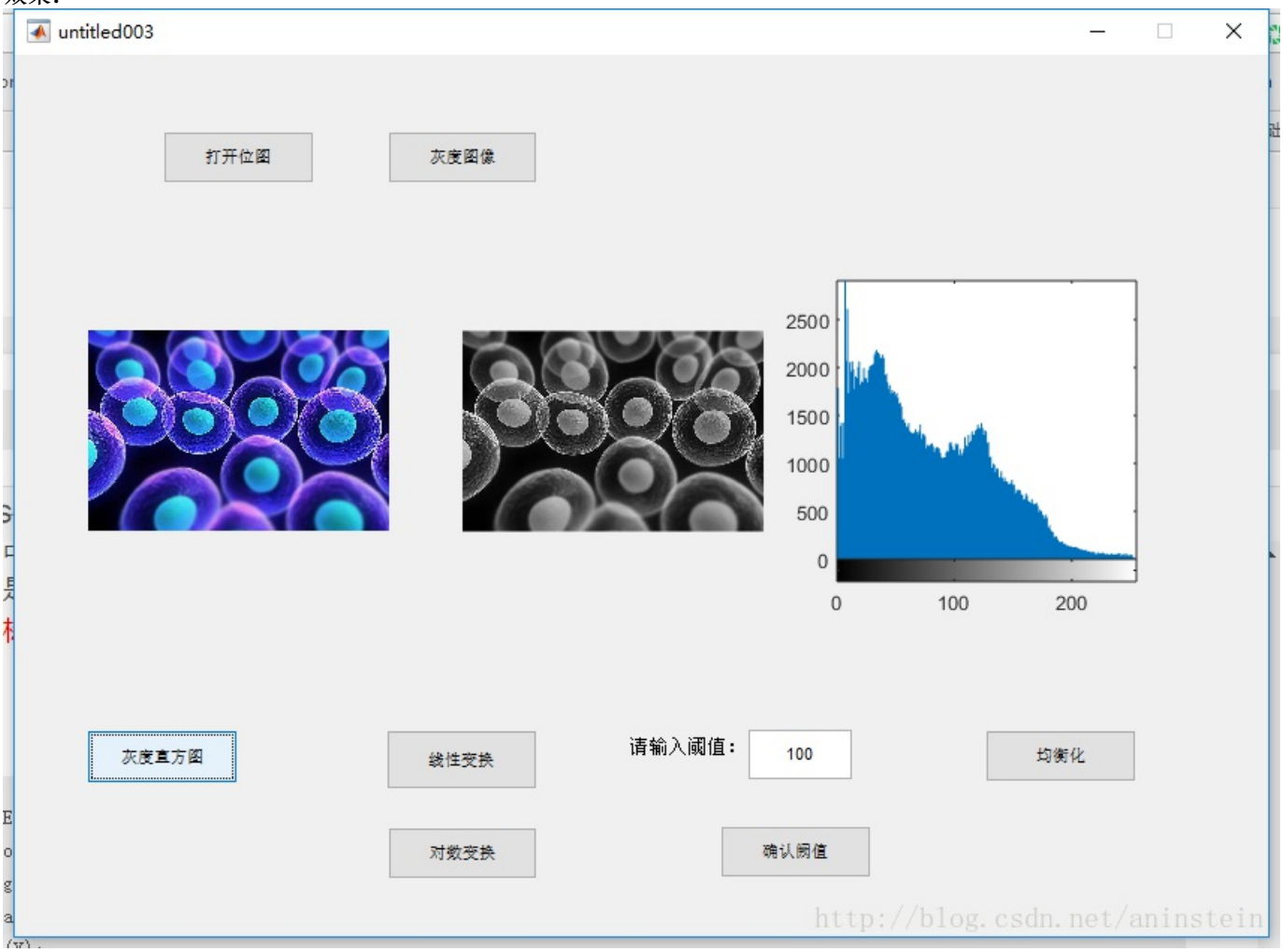
1．实现灰度直方图。

说明：一幅图像由不同灰度值的像素组成，图像中灰度的分布情况是该图像的一个重要特征。图像的灰度直方图就描述了图像中灰度分布情况，能够很直观的展示出图像中各个灰度级所占的多少。 图像的灰度直方图是灰度级的函数，描述的是图像中具有该灰度级的像素的个数：
其中，横坐标是灰度级，纵坐标是该灰度级出现的频率。

代码：

```
% --- Executes on button press in 显示灰度直方图.
function grayBar_Callback(hObject, eventdata, handles)
X=rgb2gray(handles.I);
axes(handles.axes2);
imhist(X);
```

效果：



2．实现灰度线性变换。

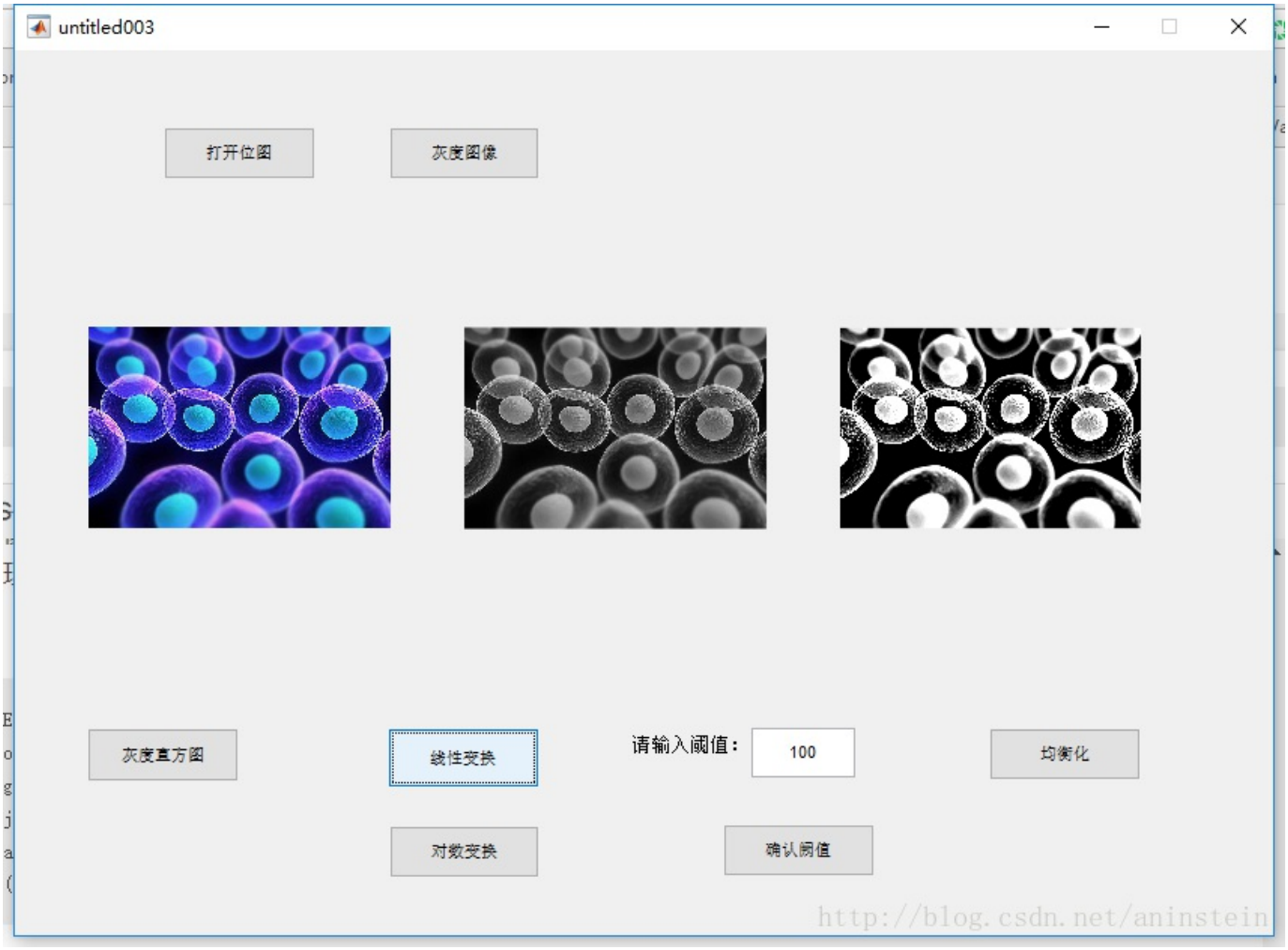
说明：灰度图像主要针对独立的像素点进行处理，由输入像素点的灰度值来决定相应的输出像素点的灰度值，通过改变原始图像数据所占据的灰度范围而使图像在视觉上得到改观，由于灰度变换没有利用像素点之间的相互关系，因而这种处理方法也叫点运算。灰度变换法又分为线性变换和非线性变换，是根据他们采用的算法来定义的。

代码：

```
% --- Executes on button press in 线性变换.
function lineTransfrom_Callback(hObject, eventdata, handles)
X=rgb2gray(handles.I);
J=imadjust(X,[0.2 0.5],[0 1]);
axes(handles.axes2);
imshow(uint8(J));
```

这里的线性变换我偷懒直接取了（0.2,0.5）到（0,1）两个坐标画直线作为变换规则，但是也是线性对吧~

仔细的可以参考：matlab图像处理-线性变换和直方图均衡化



3．实现灰度非线性变换。

说明：灰度非线性变换：

（1）负相变换

负相变换也叫做反相变换，即对每一个像素值求反。对图像求反就是讲原图的灰度值反转，简单的说就是黑的变成白的，白的变成黑的。

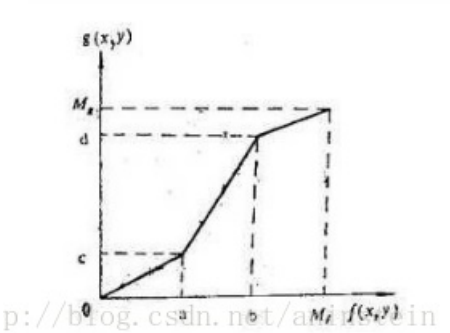
对于灰度图像或彩色图像的每个通道，其算法为：g(x,y) = 255 - f(x,y)

（2）二值化和阈值处理

二值化是分段线性的一个特例，一副图像包括目标、背景和噪声，怎样从多值的灰度图像中提取出目标？最常用的方法就是设定一个阈值θ，用θ将图像的数据分成两部分：大于θ的像素群和小于θ的像素群。例如，输入图像为f(x,y)，输出图像为f'(x,y),则f'(x,y) = 255（f(x,y) >= θ）；f'(x,y) = 0（f(x,y) <= θ），这就是图像二值化处理，它的目标就是利用一个阈值θ将图像f(x,y)分成目标和背景两个领域。

（3）分段线性变换

分段线性变换也叫做灰度线性拉伸，常用的是分三段分线性变换。如下图：



图中对灰度区间[a,b]进行了扩展，而灰度区间[0, a]和[b, Mf]收到了压缩。通过细心调整折线拐点的位置及控制分段直线的斜率，可对任意灰度区间进行扩展和压缩。、

对于非线性灰度变化，只是算法实现和最终效果上有区别，与线性变换一样，都是为了改善图像的质量。

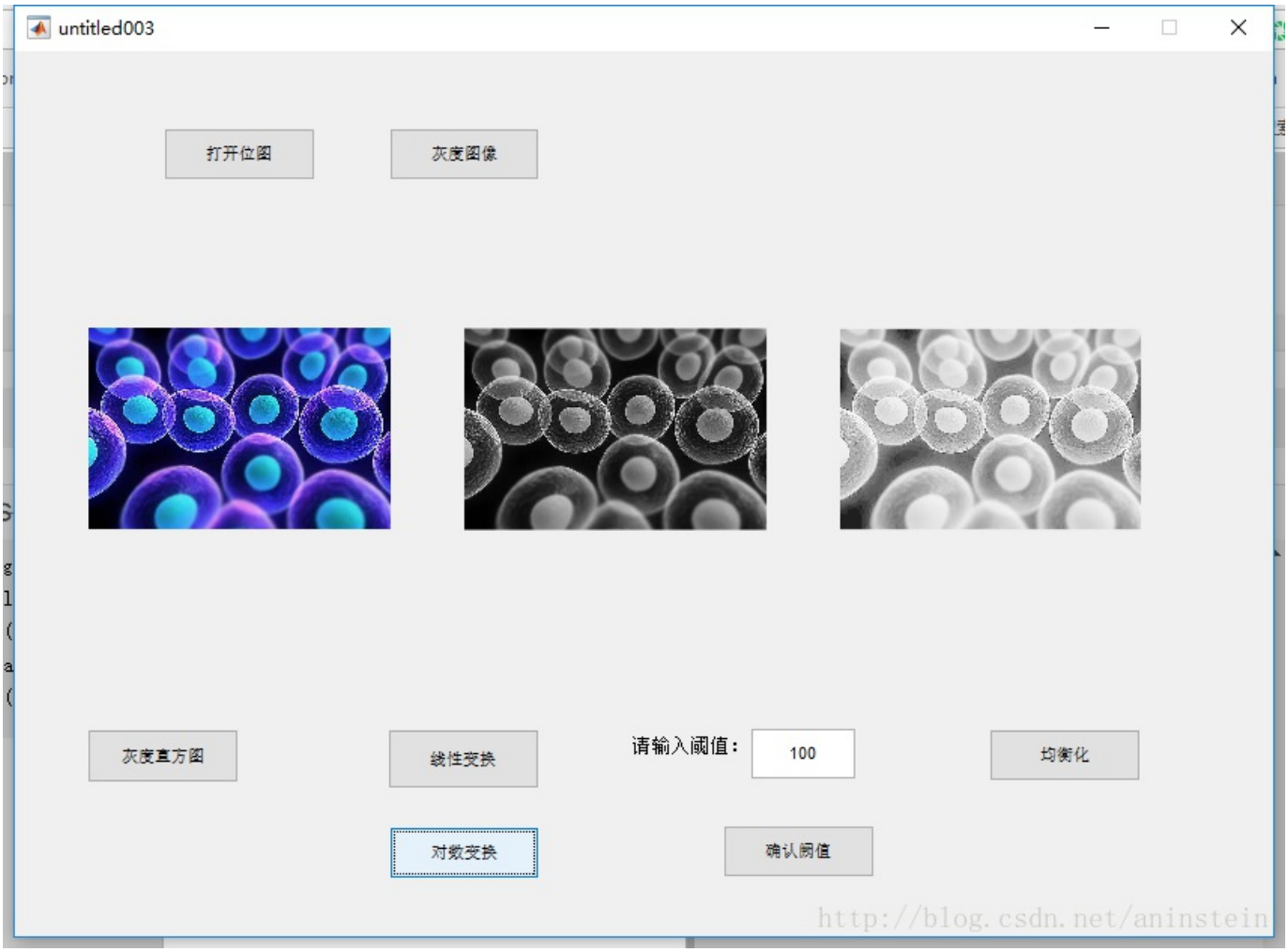
（4）对数变换，顾名思义就是以对数的函数图像作为变换规则。

这里只演示对数图像变换：

代码：

```
% --- Executes on button press in 对数变换.
function logTransfrom_Callback(hObject, eventdata, handles)
X=rgb2gray(handles.I);
J=double(X);
H=(log(J+1))/10;
axes(handles.axes2);
imshow(H,handles.map);
```

效果：



4．实现阈值变换。

阈值变换：灰度的阀值变换可以将一幅灰度图像转换成黑白二值图像。操作过程如下：先由用户指定一个阀值，如果图像中某像素的灰度值小于该阀值，则将该像素的灰度值设置为0，否则灰度值设置为255。

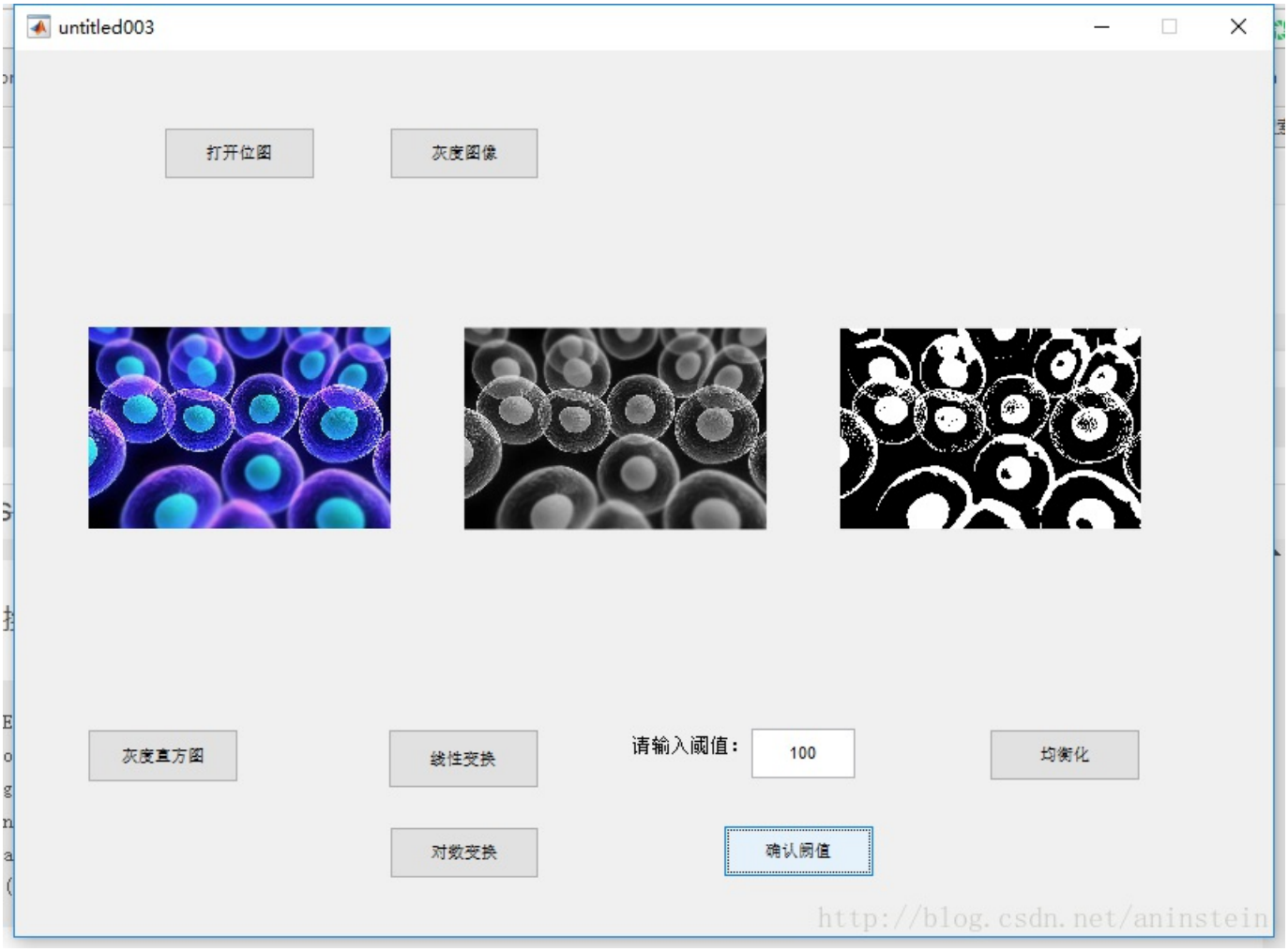
从编辑框输入阈值的代码：

```
% 阈值输入
function threshold_Callback(hObject, eventdata, handles)
handles.user_entry = str2double(get(hObject,'string'));% 从编辑框中取出阈值，变为数值型
if isnan(handles.user_entry)
    errordlg('enter a numeric value','Bad Input','modal')
    uicontrol(hObject)
    return
end
guidata(hObject,handles)
```

进行阈值变换的代码：

```
% --- Executes on button press in 确认阈值并且进行阈值变换.
function confirmThreshold_Callback(hObject, eventdata, handles)
X=rgb2gray(handles.I);
J=X>handles.user_entry;
axes(handles.axes2);
imshow(J,handles.map);
```

效果：



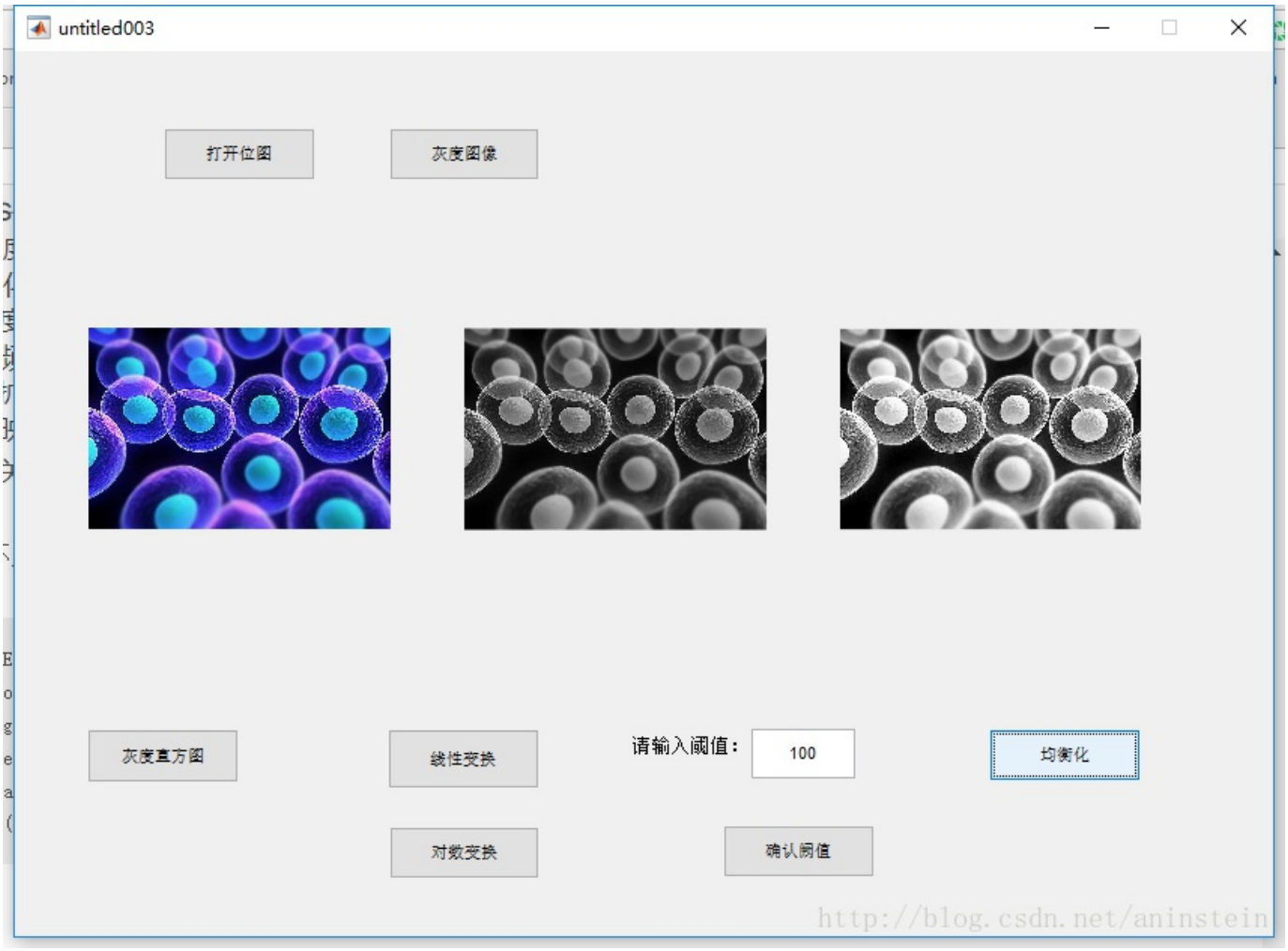
5．实现均衡变换。

说明：直方图均衡主要用于增强动态范围偏小的图像的反差，其基本思想是把原始图像的直方图变换为均匀分布，从而增强灰度值的动态范围，以达到增强对比度的效果。
直方图均衡化算法如下：
1.归一化灰度频数直方图，得到频率直方图sk
2.用sk计算频率累计直方图tk，
3.tk做取整扩展：tk = int[(L - 1) * tk + 0.5]，将直方图灰度映射尽量满整个灰度取值空间L
4.确定变换映射关系k->tk
5.根据映射关系变换图像灰度值

当然我们都不用管，直接上代码就可以：

```
% --- Executes on button press in 均衡化.
function Equalization_Callback(hObject, eventdata, handles)
X=rgb2gray(handles.I);
J=histeq(X);%直方图均衡化函数
axes(handles.axes2);
imshow(uint8(J));
```

效果：


untitled003
打开位图
灰度图像
灰度直方图
线性变换
请输入阈值：
100
均衡化
对数变换
确认阈值
http://blog.csdn.net/aninstein