《数字图像处理》

期末大作业

题 目：基于图像处理的自动报警系统

学 号：

姓 名：

2018年12月

[1GUI的设计 1](#_Toc532830058)

[1.1 GUI设计的意义 1](#_Toc532830059)

[1.2 页面设计 1](#_Toc532830060)

[1.2.1具体操作 1](#_Toc532830061)

[1.3页面显示结果 2](#_Toc532830062)

[2读取图像并显示出来图像 2](#_Toc532830063)

[2.1算法原理 2](#_Toc532830064)

[2.2算法设计 2](#_Toc532830065)

[2.3实验结果及分析 3](#_Toc532830066)

[3对图像进行灰度化处理 4](#_Toc532830067)

[3.1算法原理 4](#_Toc532830068)

[3.2算法设计 4](#_Toc532830069)

[3.3实验结果及分析 5](#_Toc532830070)

[4图像中值滤波处理 5](#_Toc532830071)

[4.1算法原理 5](#_Toc532830072)

[4.2算法设计 6](#_Toc532830073)

[4.3实验结果及分析 6](#_Toc532830074)

[5二值化处理 7](#_Toc532830075)

[5.1算法原理 7](#_Toc532830076)

[5.2算法设计 7](#_Toc532830077)

[5.3实验结果及分析 9](#_Toc532830078)

[6检测是否有人进入 10](#_Toc532830079)

[6.1算法原理 10](#_Toc532830080)

[6.2算法设计 10](#_Toc532830081)

[6.3实验结果与分析 10](#_Toc532830082)

[7形态学处理 12](#_Toc532830083)

[7.1算法原理 12](#_Toc532830084)

[7.2算法设计 12](#_Toc532830085)

[7.3实验结果与分析 13](#_Toc532830086)

[8结论 14](#_Toc532830087)

[9心得体会 14](#_Toc532830088)

[10参考文献 14](#_Toc532830089)

# 1GUI的设计

## GUI设计的意义

通常在开发一个实际应用的程序时会尽量做到界面友好， 最常使用的方法就是使用图形用户界面 GUI。建立这样一个界面友好、占用资源少、高性能、便于移植、可配置的 GUI界面设计，能够使用户的学习和使用更为方便容易。用户不需要知道后台的应用程序究竟是怎样执行各种命令的，而只需了解可见界面组件的使用方法； 只要通过与界面交流就可以使指定的行为得到正确的执行，对输入的信息进行一系列的处理。设计GUI界面，能够方便直观地对参数选择很好的表达出来。及时观察参数变化过程中的正则化参数的变化选取， 对于研究一些信号、图形、图像处理技术具有十分重要的参考意义。

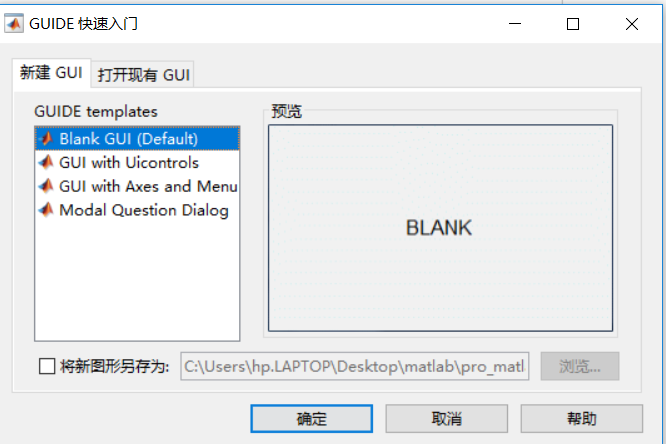
## 1.2 页面设计

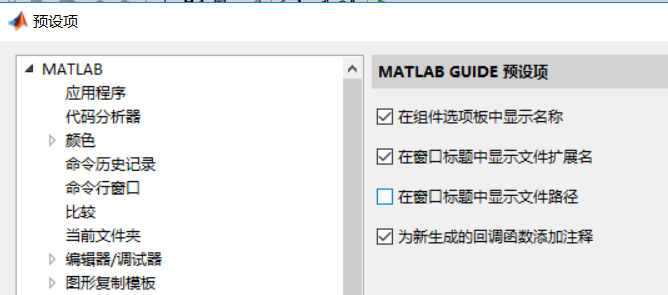
新建一个.m脚本，并且在这个脚本上输入guide,出现创建页面的界面，选择空白页面，在GUI界面中的文件预设里，勾上在组件选项板中显示名称。便于了解是什么样的组件。

直接拖拽左方的组件到右方的页面上，双击组件可进入到组件的属性栏中，对其进行内容编辑，其中Tag是标签，一般不进行改动。

在页面的最上方选择一个文本框，用来显示自己的学号和姓名，由实验内容分析可以得到，首先要得到原无人进入的原图像以及需要检测的图像，设计了两个按钮用来实现，要对图像进行灰度化，中值滤波进行平滑，以及二值化处理，因为二值化处理是要选择不同的阈值，故选择下拉菜单的形式，可以选择一个具体的阈值，而不是设计很多的按钮。还有判断是否有人进入，以及对结果进行一系列的形态学操作。在这里具体的功能都选择用的是按钮进行操作。

1.2.1具体操作



## 1.3页面显示结果



# 2读取图像并显示出来图像

## 2.1算法原理

图像的显示过程是将数学图像从一组离散数据还原为一幅可见图像的过程，通过图像显示，可以直观地查看和验证我们对图像的操作是否正确。其中读取图像用到了Matlab函数库里的一个函数。其中，FileName：返回的文件名PathName：返回的文件的路径名FilterIndex：选择的文件类型FilterSpec：文件类型设置DialogTitle：打开对话框的标题DefaultName：默认指向的文件名。

当读取文件以后，返回文件的详细路径，用;用来读取这个图片，用;用来显示这个图像。

## 2.2算法设计

选择图片格式为.jpg格式的图片，格式可进行更改，并不一定非要是jpg格式的。在选择完图片以后要把图片的具体路径和名称传出来，便于以后对于图片的一系列的操作。

function [ str] = readPicture( )

%READPICTURE 此处显示有关此函数的摘要

% 选择图片并将其显示出来

%选择图片路径

[filename,pathname,filterindex]=uigetfile({'\*.jpg';'\*.bmp';'\*.gif'},'选择图片')%选择图片是窗口的名字

%合成路径+文件名

str=[pathname filename];

end

main.m:

根据函数的返回值，得到图片的具体路径和名称，并且把图片显示出来

str=readPicture();

global I;

I=imread(str);

subplot(1,2,1)

imshow(I);

title('原始图像');

## 2.3实验结果及分析



**分析：**这里把图片设为全局是因为，当没有设置成全局时，发现在后续的操作对这个图片进行操作的时候，显示这个图片并不存在，这时了解到如果并没有加全局修饰符时，相当于是一个函数里的变量，作用域只写那个函数的内部，当在函数外使用时，并没有其值，所以把图片设置为全局变量。

# 3对图像进行灰度化处理

## 3.1算法原理

将彩色图转化成为灰度图的过程称为图像的灰度化处理。 彩色图像中的每个像素的颜

色有 R、G、B 三个分量决定，而每个分量有 255 个中值可取，这样一个像素点可以有 1600多万（ 255\*255\*255）的颜色的变化范围。而灰度图像是 R、G、B 三个分量相同的一种特殊的彩色图像， 其中一个像素点的变化范围为 255种，所以在数字图像处理中一般先将各种格式的图像转变成灰度图像以使后续的图像的计算量变得少一些。 灰度图像的描述与 彩色图像一样仍然反映了整幅图像的整体和局部的色度和亮度等级的分布和特征。

在 RGB 模型中，如果 R=G=B 时，则彩色表示一种灰度颜色， 其中 R=G=B 的值叫做

灰度值。因此，灰度图像每个像素只需一个字节存放灰度值（又称强度值、亮度值） ，灰度范围为 0-255。在Matlab工具箱中有将彩色图像变灰度化的函数。

## 3.2算法设计

在灰度化处理按钮的callback(回调)的下方输入这些代码，对原图像以及需要检测的图像进行灰度化的处理。

global I;%表示的是原图像

global I1;%表示的是用于检测的图像

I=rgb2gray(I);%将原图像进行灰度化处理

%figure('NumberTitle', 'off', 'Name', '处理过程');%显示图片新建一个窗口用来显示

subplot(1,2,1);

imshow(I);%显示原图像的灰度化

title('原始图像灰度化');

subplot(1,2,2);

I1=rgb2gray(I1);%将检测的图像灰度化

imshow(I1);%显示检测图像的灰度化

title('目标图像灰度化');

## 3.3实验结果及分析



**分析**：对图片进行灰度化处理，便于后续对图片的一系列的操作。而不是得到图像的R分量灰度图，G分量灰度图，B分量灰度图，对三个图像进行处理，在进行合并。以及当设置一个变量为全局变量，在用这个变量时，也要先进行global I，才可以继续使用这个全局变量。

# 4图像中值滤波处理

## 4.1算法原理

中值滤波法是一种非线性平滑技术，它将每一像素点的灰度值设置为该点某邻域窗口内的所有像素点灰度值的中值。

中值滤波是基于排序统计理论的一种能有效抑制噪声的非线性信号处理技术，中值滤波的基本原理是把数字图像或数字序列中一点的值用该点的一个邻域中各点值的中值代替，让周围的像素值接近的真实值，从而消除孤立的噪声点。

方法是用某种结构的二维滑动模板，将板内像素按照像素值的大小进行排序，生成单调上升（或下降）的为二维数据序列。二维中值滤波输出为g（x,y）=med{f(x-k,y-l),(k,l∈W)} ，其中，f(x,y)，g(x,y)分别为原始图像和处理后图像。W为二维模板，通常为3\*3，5\*5区域，也可以是不同的的形状，如线状，圆形，十字形，圆环形等。

中值滤波使用的是Matlab函数库里的函数，;注使用这个函数，图像应当为二维图像，普通的RGB为三维图像，因在这个函数之前已经调用了灰度化函数，故不存在图像为三维的问题。

## 4.2算法设计

在图像中值化处理的按钮下输入下方的代码，用于对原始图像和检测图像进行中值滤波。

global I;

global I1;

I=medfilt2(I,[5,5]);%对原图像进行中值滤波

I1=medfilt2(I1,[5,5]);%对检测图像进行中值滤波

subplot(1,2,1);

imshow(I);%显示原图像

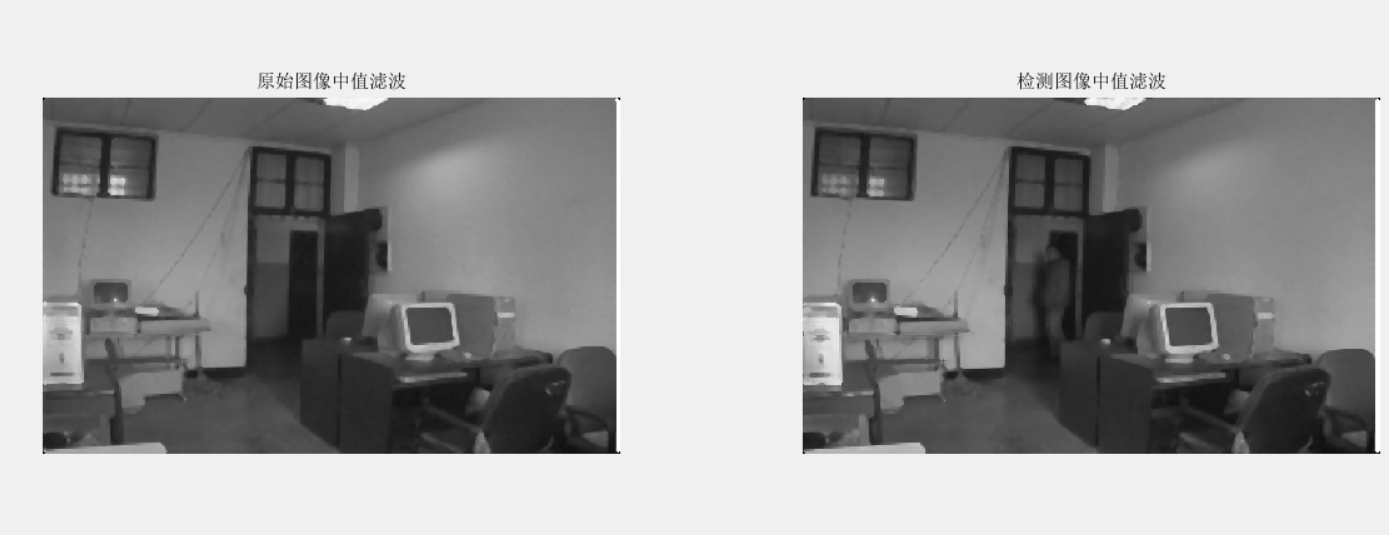
title('原始图像中值滤波');

subplot(1,2,2);

imshow(I1);%显示检测图像

title('检测图像中值滤波');

## 4.3实验结果及分析



**分析**：中值滤波函数可对图像进行一定的降噪处理，并且图像经过中值滤波后，高斯噪声没有被完全去除，椒盐噪声几乎被完全去除效果较好。

# 5二值化处理

## 5.1算法原理

图像的二值化处理就是将图像上的点的灰度值为0或255，也就是将整个图像呈现出明显的黑白效果。即将256个亮度等级的灰度图像通过适当的阈值选取而获得仍然可以反映图像整体和局部特征的二值化图像。在数字图像处理中，二值图像占有非常重要的地位，特别是在实用的图像处理中，以二值图像处理实现而构成的系统是很多的，要进行二值图像的处理与分析，首先要把灰度图像二值化，得到二值化图像，这样子有利于在对图像做进一步处理时，图像的集合性质只与像素值为0或255的点的位置有关，不再涉及像素的多级值，使处理变得简单，而且数据的处理和压缩量小。为了得到理想的二值图像，一般采用封闭、连通的边界定义不交叠的区域。所有灰度大于或等于阈值的像素被判定为属于特定物体，其灰度值为255表示，否则这些像素点被排除在物体区域以外，灰度值为0，表示背景或者例外的物体区域。

如果某特定物体在内部有均匀一致的灰度值，并且其处在一个具有其他等级灰度值的均匀背景下，使用阈值法就可以得到比较的分割效果。如果物体同背景的差别表现不在灰度值上（比如纹理不同），可以将这个差别特征转换为灰度的差别，然后利用阈值选取技术来分割该图像。动态调节阈值实现图像的二值化可动态观察其分割图像的具体结果。

## 5.2算法设计

在下拉列表的callback下方输入了具体的二值化操作，获取下拉列表的值，得到二值化的阈值。

%对下拉菜单进行的一系列操作

% --- Executes on selection change in popupmenu1.

function popupmenu1\_Callback(hObject, eventdata, handles)

% hObject handle to popupmenu1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

val=get(handles.popupmenu1,'value');

disp(val);%输出的是选择的第几个而不是他的具体内容

%disp函数显示文本

switch(val)

case 2

T=40;

case 3

T=50;

case 4

T=60;

end

disp(T);%输出文本

%原始图像

global I;

%检测图像

global I1;

%对原始图像进行二值化处理

I=im2bw(I,T/255);

I1=im2bw(I1,T/255);

subplot(1,2,1);

imshow(I);

title('原始图像二值化处理');

subplot(1,2,2);

imshow(I1);

title('检测图像二值化处理');

## 5.3实验结果及分析







**分析**：当二值化的阈值选择60时，目标与干扰信息的区别不大，很难很好的检测到是否有人进入。以为课题要求阈值选取应当为40 50 60，弄按钮不是很好，故选用下拉列表的形式存放阈值。

# 6检测是否有人进入

## 6.1算法原理

利用背景相减法对运动目标进行检测该算法就是将待处理图像与标准图像的对应像素值相减，完成对是否有人进入的监控。在这里使用的是两个图像的二值化图像进行相减，二值化图像的像素值只有0与255，容易去除更多的扰乱信息，在这里使用的函数是Matlab函数库里的函数，对两个图片进行相减。 ;

## 6.2算法设计

%原始图像

global I;

%检测图像

global I1;

I=imsubtract(I1,I);

subplot(1,1,1);

imshow(I);

title('两张图片进行相减');

## 6.3实验结果与分析







分析：检测是否有人进入的图像相减受到了二值化处理的影响，可看出当阈值选择60时，图像相减时，目标区域与干扰区域的差别不大。减法运算增强了图像间的差别。检测了同一场景的两幅图像之间的变化。

# 7形态学处理

## 7.1算法原理

因为有干扰区域的干扰，所以要对两个图片相减的结果进行形态学的处理。一般，结构元素的尺寸要明显小于目标图像的尺寸。腐蚀可以消除小且无意义的目标物（毛刺、 小凸起)。若一个目标区域中含有细小的连接部分，可以选取足够大的结构元素，通过腐蚀运算将其腐蚀掉实现区域分割。**腐蚀的原理**：拿结构元素S的原点和目标图像X上的点一个一个地对比，如果S上的所有点都在X的范围，则S的原点对应的点保留，否则将该点去掉。**膨胀的原理**：拿结构元素S的原点和目标图像X上的点及X周围的点一个一个地比对，如果S上有一个点落在X的范围内，则该点存在且为黑。

膨胀用到了MATLAB图像处理工具箱提供用于图像膨胀的函数是

腐蚀用到了MATLAB图像处理工具箱提供用于图像腐蚀的函数是

## 7.2算法设计

先清除边界图像，在腐蚀掉与目标区域差别很大的干扰区域，在对目标区域进行膨胀。

global I;

I=imclearborder(I,4);

subplot(1,3,1);

imshow(I);

title('清除边界图像');

se = strel('disk',4);

global erodedBW;

erodedBW = imerode(I,se);%对图像进行腐蚀

subplot(1,3,2);

imshow(erodedBW);

se1 = strel('line',1,0);

erodedBW1 = imerode(erodedBW,se1);%在对图像进行腐蚀

subplot(1,3,2);

imshow(erodedBW1);

title('腐蚀运算结果');

global dilatedBW;

dilatedBW = imdilate(erodedBW,se)%对图像进行膨胀;

subplot(1,3,3);

imshow(dilatedBW);

title('膨胀运算结果');

## 7.3实验结果与分析



**分析**：当阈值为60 时，由于目标区域与干扰区域之间的差距不大，导致了进行形态学处理时，不能很好的只留下目标区域。

# 8结论

对Matlab页面的GUI不是十分的了解，并没有做出菜单栏的样式，只做出来简单的按钮以及下拉列表的样式。以及自己自定义的函数只有一个，大多都使用的是Matlab函数库里的函数，以及对于最后一步形态学的处理并不是很好，当进行中值滤波的时候，当阈值为40 以及50的时候，形态学处理虽然有小杂物，但是可具体看的出来是否有人入侵，当阈值为60时形态学的处理效果不好，因为目标物与杂物之间的差异不是很大，无法很好的进行腐蚀，改进之处：应该在进行选择中值滤波的阈值时，应将阈值的值保存下来，在形态学处理时根据不同的阈值进行不同的形态学处理。

# 9心得体会

经过这次对基于图像处理的自动报警系统的处理，对Matlab的GUI有了一些了解，以及对自己的设计思路也有了一定的提高，当拿到一个题目时，不应该直接就上手对它进行编程，应该在设计之前就应该想好怎么弄，避免弄到一半，出现问题。以及对Matlab对于图像的基本处理有了一定的自己的理解，不在是只会根据提示怎么弄，更多的有了自己的想法，在这个课题中，运用了图像的灰度化处理，以及对图像的中值滤波处理，对图像进行二值化，以及一些形态学处理。在做这个课题之前，自己对于形态学的了解并不是很懂，经过做这个课题，对形态学的处理有了自己的处理，对于一个图像是要进行开运算和闭运算有了自己的判断

# 10参考文献