

2016 级英才实验学院信号与系统  
( Course of signals and systems  
for the class of 2016  
of Yingcai Honors College of UESTC )  
课程设计报告

设计题目：模拟手机 APP 美化照片——基于  
MATLAB 的图像处理实验

设计人员：杨鹏志 2016040203030  
李映泉 2016030401019

日 期：2017 年 12 月 24 日

设计人员班级：2016000202 班

# 信号与系统课程设计 计结题报告

杨鹏志 2016040203030

李映泉 2016030401019

## 一、负责内容

滤镜部分由李映泉负责，主要为两大部分：滤镜程序编写和频域性质展示。其中滤镜程序含四个部分：素描滤镜、浮雕滤镜、毛玻璃滤镜、颜色重组滤镜。

GUI 部分由杨鹏志负责，整体架构为三个部分：界面搭建、对象浏览器属性设置、回调函数的编写。

## 二、滤镜部分

### 1. 方案与原理

#### 1.1. 操作环境及输入输出设置

##### 1.1.1. 操作环境

本设计的整体仿真环境为 MATLAB R2014b，在 GUI 导入图片，再通过 GUI 输出经过处理后的图片，为节省时间限制图片大小为 1000\*1000\*50%像素，采用的算法有：去色、反色、高斯模糊、颜色减淡、卷积、随机 RGB 值替换、RGB 权重重调、傅立叶变换。

#### 2.1.2. 图片输入设置

利用 `imread()` 函数读入图片，下面为叙述方便统一读入以下图片，格式为 JPEG



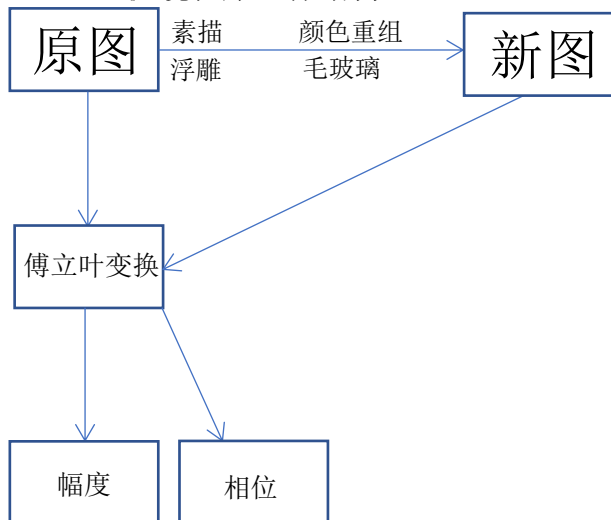
Figure 1

具体代码实现见附录。规定读入尺寸为 1000\*1000 像素，输出时虽然只按照 50% 输出但是不影响效果。

#### 2.1.3. 图片输出设置

所有输出结果均通过图片形式展现，使用 `figure()`，`imshow()` 函数和 `title()` 函数实现输出。具体代码见附录。

#### 2.2. 滤镜程序整体结构



#### 2.3. 滤镜

##### 2.3.1. 素描滤镜

输入图片经过素描滤镜后可以输出铅笔画的风格。仿佛素描版的原图片。

依次涉及的主要算法有：去色、反色、高斯模糊、颜色减淡。

去色：用灰度表示图片，使用 `rgb2gray()` 函数实现。

反色：用 255 减去图片上任意一点的灰度值，即取当前色关于中性灰的相反色。设当前点灰度值为  $x$ ，则反色后该点灰度为  $255-x$ 。

高斯模糊：减少图像噪声以及降低细节层次，使图片看起来平滑。设当前点为  $(i, j)$ ，灰度值为  $N(i, j)$ ，设定中间变量  $sum$ ，则先将

$$N(i-1, j-1) + 2 * N(i-1, j) + N(i-1, j+1)$$

赋给  $sum$ ，再将

$$sum + 2 * N(i, j-1) + 4 * N(i, j) + 2 * N(i, j+1)$$

赋给  $sum$ ，最后将

$$sum + N(i+1, j-1) + 2 * N(i+1, j) + N(i+1, j+1)$$

赋给 sum，最后将 sum/16 的值设为当前点灰度值。

颜色减淡：查看每个通道中的颜色信息，并通过降低对比度使基色变亮以反映混合色，由混合色的亮度决定基色的亮度和反差。任何颜色与白色复合产生白色，任何颜色与黑色复合保持不变。给定目标像素点灰度值 b，源像素点灰度值 a，则最终灰度值  $c = a + a * b / (256 - b)$ ，为避免色阶溢出，使用 min() 函数选择 c 与 255 中较小值。

### 2.3.2. 浮雕滤镜

输入图片经过浮雕滤镜后可以输出浮雕的风格。仿佛将原图片用浮雕艺术展现。

涉及的主要算法有：去色、卷积。

去色不再赘述，卷积是将当前点的灰度值用相邻点的灰度值表示，具体是：设当前点灰度值是  $N(i, j)$ ，则卷积后灰度值为

$$g(i, j) = N(i-1, j-1) - N(i+1, j+1) + 128$$

### 2.3.3. 毛玻璃滤镜

输入图片经过毛玻璃滤镜后可以输出毛玻璃的风格。仿佛透过毛玻璃观察原图片。

涉及的主要算法有：随机 RGB 值替换。为作出毛玻璃风格需要将原图模糊且是无取向模糊，这样图片看上去平滑且仅接近透过毛玻璃看到的效果。具体是：使用 rand() 产生随机数，并设置半径常数 N，则当前点  $(i, j)$  的 RGB 值为  $(i-1, j-1)$  处 RGB 值与  $(i-[N * \text{rand}()], j-[N * \text{rand}()] )$  ( $[]$  代表向下取整) 处 RGB 值之和，若  $i-[N * \text{rand}()]$  或  $j-[N * \text{rand}()]$  小于 0 则取  $i-[N * \text{rand}()] = [N * \text{rand}()]$ ，即再次生成一随机数代替负值。

### 2.3.4. 颜色重组滤镜

输入图片经过颜色重组滤镜后可以改变色彩冷暖度。仿佛图片重新上色一样。

涉及的主要算法有：RGB 权重重调。即重新设置红、绿、蓝三种颜色的比重。设原图中红色分量为 R，绿色分量

为 G，蓝色分量为 B，设置参数 a, b, c，令  $j = 3 / (a + b + c)$ ，则经过滤镜后图片的红、绿、蓝三分量依次为  $R' = a * j * R$ ， $G' = b * j * G$ ， $B' = c * j * B$ 。使用 imshow() 输出即可。

### 2.4. 频域特性

使用二维傅立叶变换得到图片的频域特性，再使用 imshow() 输出即可，MATLAB 中 fft() 函数可以实现二维傅里叶变换。变换后分别使用 abs() 和 angle() 求出幅度和相角，就得到图片的频域特性。为减小数值跨度，对幅度值取对数，最初输出  $\log(\text{abs}() + 1)$ ，(+1 是为保证真数为正)，观察经过滤镜前后的幅度和相角可以看出滤镜对频域特性的影响。

## 3. 结果

### 3.1. 素描滤镜

经过素描滤镜得到如下图片



Figure 2

### 3.2. 浮雕滤镜

经过浮雕滤镜得到如下图片

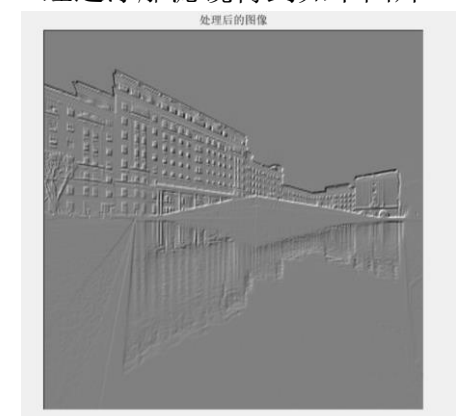


Figure 3

### 3.3. 毛玻璃滤镜

经过毛玻璃滤镜得到如下图片



Figure 4

### 3.4. 颜色重组滤镜

经过颜色重组滤镜得到如下图片



Figure 5

### 3.5. 傅立叶变换

#### 3.5.1. 原图频域特性:

通过傅立叶变换得到原图的频域特性如下:

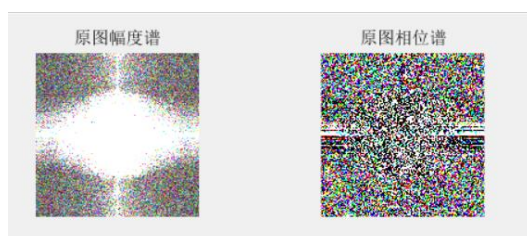


Figure 6

#### 3.5.2. 素描滤镜处理后的频域特性:

通过傅立叶变换得到经过素描滤镜处理后的图片的频域特性如右上角

所示:

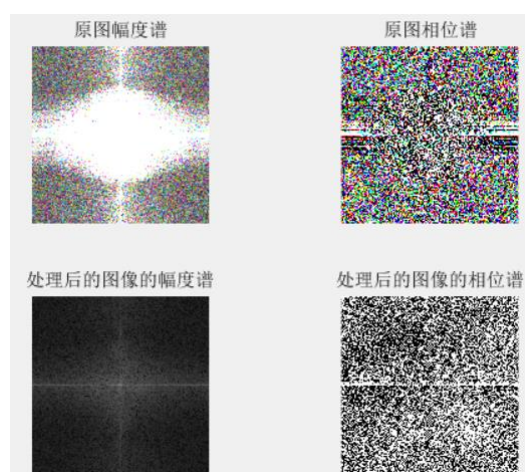


Figure 7

#### 3.5.3. 浮雕滤镜处理后的频域特性:

通过傅立叶变换得到经过浮雕滤镜处理后的图片的频域特性如下:

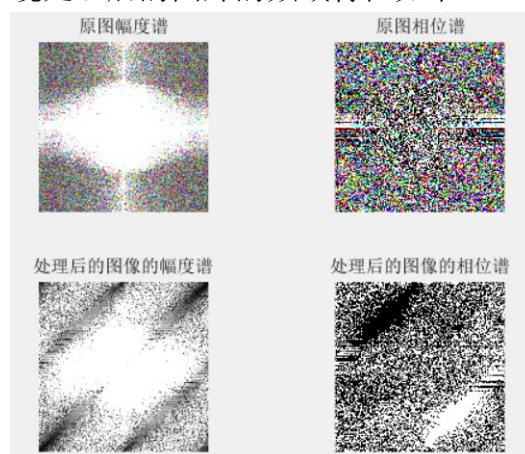


Figure 8

#### 3.5.4. 毛玻璃滤镜处理后的频域特性:

通过傅立叶变换得到经过毛玻璃滤镜处理后的图片的频域特性如下:

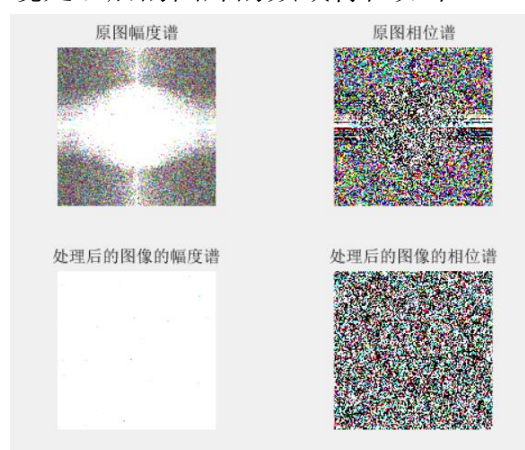


Figure 9

### 3.5.5. 颜色重组滤镜处理后的频域特性：

通过傅立叶变换得到经过颜色重组滤镜处理后的图片的频域特性如下：

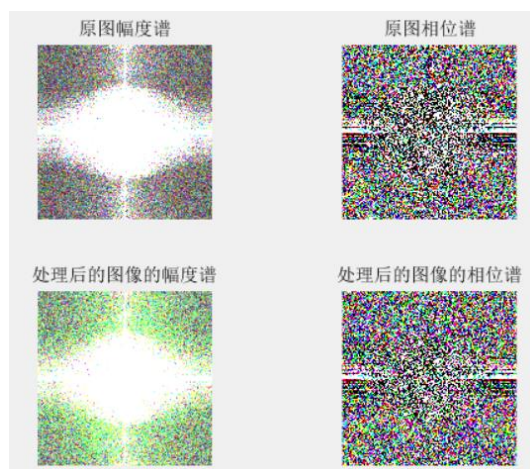


Figure 10

### 4. 不足之处

虽然根据编写的程序的确得出了良好的图片处理效果，但是还存在以下三点不足：

(1) 程序允许时间长。每一个滤镜程序大约需要 5-6 秒的时间才能运行完成。

(2) 程序的处理效果不完美。虽然经过系统后输出的图片和经过实际滤镜处理后的图片较为相似但是契合度还有提高的空间。

(3) 程序较为简单，能实现的功能少，人机交互不易。

### 4. 结论

基于 MATLAB R2014b 仿真了素描滤镜、浮雕滤镜、毛玻璃滤镜和颜色重组滤镜，并观察了经过滤镜前后的频域特性的变化，对设计的滤镜系统的工作原理做了阐释，并总结了存在的不足，指明了将来的改进方向。通过本次课程设计，加强了对于信号与系统课程内容的理解，对于什么是信号和什么是系统有了清楚的答案，课程设计圆满结束。

## 三、GUI 部分

## 1. 综述

### (1) 界面搭建

整个界面为 GUI 的基础，也是用户面对的内容。我们课设的主题是制作多种滤镜，简单地重现 PS 的一些重要功能，但是单纯的滤镜会使不擅长软件的使用者陷入困境，故搭建一个操控界面可以使问题直观化。本部分只是单纯地将各种功能的控件拖入设计页面即可。

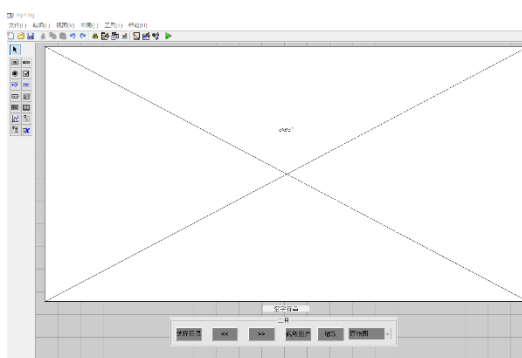


Figure 11

### (2) 对象浏览器属性设置

双击设计页面的控件，就可以打开各自的检查器，在其中可以设置非常多的属性例如字体、颜色等，在对象浏览器中即可找到各个控件。这一步起到了桥梁的作用。

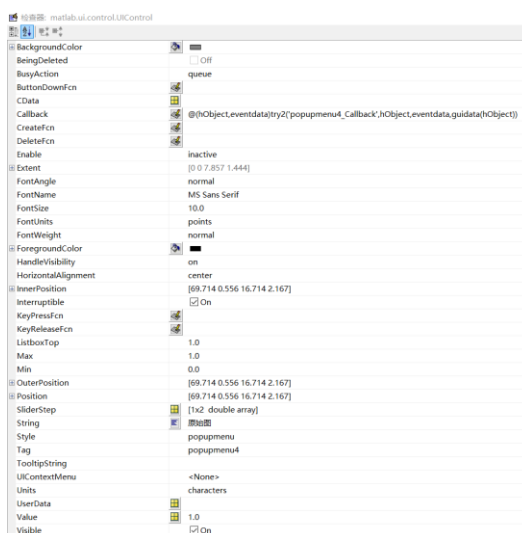


Figure 12

### (3) 回调函数编写

这是本问题的最核心内容也是难点。



每一个控件都可以设置回调函数，并且针对不同的功能有不同的方法，在下一点中详细介绍。

## 2. 难点突破

已经提到，本部分最难之处在于如何编写回调函数，自己经验较少，其中多次遇到困难。

### （1）调入照片

一般采用文件调入的方式：直接调入图片或者选择文件夹调入，为了浏览照片的方便，我们最后敲定了后者（但是最后也因此造成了一个很大的麻烦）。

### （2）与滤镜的主题相联系

鉴于我们的主题，这一步也成为 GUI 部分的重中之重。因为 m 文件的各个函数间没有联系，如果想在函数里利用另外一个的数据就必须将那个函数的数据通过特殊方式传出，有多重方法比如全局变量、application 函数等。为了防止混淆，此处选择利用句柄的方式传递数据：

```
handles.name=M;
```

将 M 的值传入句柄语句中，并利用下一句把数据传出：

```
guidata(hObject, handles);
```

最后在各个函数中，将数据得到用到如下的语句：

```
M=handles.name;
```

结合这部分的实际工作，要传递的就是照片的数据，为了处理已调入并显示到界面上的照片，采取如上的方式传递即可。如此一来八种滤镜的封装就可以一次性地嵌入，大大地简化了工作。

### （3）bug 与 debug

#### ①找不到回调函数的问题

在检查器中针对不同的功能可以对应不同的回调函数，但是如果此处无需回调函数（即编辑器中没有相应的语

句），但是检查器却设置了回调，就会发生报错，这时只需要删掉无用的对应即可。

```
>> try2
未定义函数或变量 'figure1_CreateFcn'。

出错 gui_mainfcn (line 95)
    feval(varargin{:});

出错 try2 (line 42)
    gui_mainfcn(gui_State, varargin{:});
```

Figure 13

#### ②传统的语法问题

如果不仔细有时可能会犯非常常规的传统错误。可以明确的一点是，虽然这部分的结构比较独特，但是基本的语法没有区别，所以可以方便地根据提示找到问题并解决。

## 3. 不足与改进方向

### （1）选用文件夹的方式带来的困难

因为前期在编写调取图片及切换图片的函数时候都是选取直接打开文件夹的方法，所以在添加滤镜时只能对文件夹的第一张图片进行处理，增加了界面操作的复杂度。可以通过直接调入照片文件的方式解决这个问题；

### （2）截图功能的不足

截图本身是 GUI 部分附加的内容，连带着导出图片的功能。在截取原图过程中操作非常顺畅，但是经过滤镜处理后再截图导出就会遇到错误（但是系统不报错），应为逻辑上有不足。

## 4. 成果展示



Figure 14

#### 四、引用

- [1]罗华飞. MATLAB GUI 设计学习手记 . 北京航空航天大学出版社;
- [2]陈超. MATLAB 应用实例精讲. 电子工业出版社;
- [3]CSDN 博客及 MATLAB 中文论坛;
- [4]《数学实验方法》电子科技大学数学科学学院 中国铁道出版社 2017 年 1 月
- [5]《基于 MATLAB 的信号与系统实验指导》 甘俊英 清华大学出版社 2014 年 7 月
- [6]《signals and systems》 Alan V. Oppenheim PUBLISHING HOUSE OF ELECTRONIC INDUSTRY 2014. 1
- [7]《信号与系统 MATLAB 实践》 孟桥, 董志方, 王琼 高等教育出版社 2008 年 12 月