

# 浙江工商大學

## 课 程 设 计



题目： 课程论文模板

课程名称： 课程名

学 院： 学院名

专 业： 专业名

班 级： 班级名

学 号： 你的学号

学生姓名： 你的名字

成 绩：

二〇二三年十一月

# 课程论文模板

## 摘要

请在这里输入摘要内容.

关键词：关键词 1，关键词 2，关键词 3

目录

一、 你的一级标题	1
1.1 你的二级标题 1 . . . . .	1
1.2 你的二级标题 2 . . . . .	1
1.3 你的二级标题 3 . . . . .	2
1.4 你的二级标题 4 . . . . .	2
附录一 Matlab 代码	4

# 一、 你的一级标题

## 1.1 你的二级标题 1

在日常生活中，我们所拍摄的对象时常不会处于完全静止的状态，而由于其运动，所拍摄的照片会产生许多噪声。但是用一些简单的逆滤波方法不能很好地处理噪声需要采用约束复原的方法，维纳滤波 (最小均方误差滤波器) 复原就是一种有代表性的约束复原方法，是使原始图像  $f(x, y)$  和复原图像  $\hat{f}(x, y)$  之间均方误差最小的复原方法。

均方误差的表达式为：

$$e^2 = E \left[ (f - \hat{f})^2 \right] \tag{1.1}$$

## 1.2 你的二级标题 2

为方便理解，现将代码中部分用到的函数的作用放入下表做简单介绍：

表 1 对模糊加噪声图像进行维纳滤波复原代码中函数的介绍

函数	作用
<code>imread(filename,fmt)</code>	根据文件名 <code>filename</code> 读取灰度或彩色图像，返回的是包含图像数据的数组
<code>rgb2gray(RGB)</code>	消除图像色调和饱和度信息同时保留亮度 将 RGB 图像或彩色图转换为灰度图像
<code>im2double(I)</code>	将灰度图像 <code>I</code> 转换为双精度
<code>size(A)</code>	获取矩阵 <code>A</code> 的行数和列数
<code>fspecial(type, para)</code>	用于建立预定义的滤波算子， 其中 <code>type</code> 指定算子的类型， <code>para</code> 指定相应的参数
<code>conv2(A,B)</code>	计算矩阵 <code>A</code> 和 <code>B</code> 的卷积
<code>zeros(m,n)</code>	产生 $m \times n$ 的零矩阵
<code>imnoise(f,type,parameters)</code>	给一幅图像添加噪声， <code>f</code> 是原图像， <code>type</code> 是加入的噪声类型， <code>parameters</code> 是噪声的一些参数
<code>figure</code>	创建一个新的窗口，所有参数采用默认
<code>imshow(I)</code>	显示灰度图像 <code>I</code>
<code>title('caption')</code>	设置图像的标题 <code>caption</code>

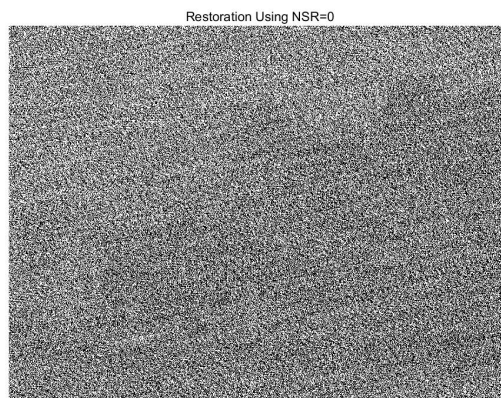
表格引用：表1.

### 1.3 你的二级标题 3

代码运行结果如下：



(a) 加噪声的模糊图像



(b) 维纳滤波结果图像

图 1 维纳滤波效果

图片引用: 图1.

### 1.4 你的二级标题 4

章节引用: 节1.4。

文献引用: [1] 或<sup>[1]</sup>。

Matlab 代码参见附录一.

## 参考文献

- [1] 蔡利梅, 王利娟. 数字图像处理. 徐州: 中国矿业大学出版社, 2014.08.

## 附录一 Matlab 代码

```
Image=im2double(rgb2gray(imread('pic2.jpg')));
window=15;
[n, m]=size(Image);
n=n+window-1;
m=m+window-1;
% 点扩散函数
h=fspecial('average', window);
BlurredI=conv2(h, Image);
% 噪声信号
noise=imnoise(zeros(n, m), 'salt & pepper', 0.001);
% 给模糊图像添加椒盐噪声
BlurrednoisyI=BlurredI+noise;
figure, imshow(BlurrednoisyI), title('Blurred Image with noise');
% 模板延拓
h1=zeros(n, m);
h1(1:window, 1:window)=h;
H=fftshift(fft2(h1));
% 计算信噪比
K=sum(noise(:).^2)/sum(Image(:).^2);
% 计算维纳滤波传递函数
M=conj(H)./(abs(H).^2+K);
G=fftshift(fft2(BlurrednoisyI));
f=ifft2(ifftshift(G.*M));
result=f(1:n-window+1, 1:m-window+1);
figure, imshow(abs(result), []), title('Filtered Image');
```