

homework1 说明

1 作业内容

1 (4 分) 参考课程第二讲 PPT, 求 λ .

2 (6 分) 用如下模型对图片进行去噪去模糊.

$$\hat{u} = \underset{u}{\operatorname{argmin}} \lambda \int |\nabla u| dx + \frac{1}{2} \int (\mathcal{A}u - f)^2 dx$$

其中 f 是待去噪去模糊的图片, u 是优化变量, 得到的最优解 \hat{u} 为处理结果. \mathcal{A} 是以 k 为卷积核的卷积算子:

$$\mathcal{A}u = k * u$$

- 卷积核 $k = \text{fspecial}(\text{'gaussian'}, [s,s], \sigma)$, 大小为 $[s,s]$ (作为输入参数待定), 方差 σ , 以 k 下标从 1 至 s 为例,

$$k_{ij}^g = e^{-\frac{(i-\frac{s+1}{2})^2 + (j-\frac{s+1}{2})^2}{\sigma^2}}, i, j = 1, \dots, s$$
$$k_{n_1 n_2} = \frac{k_{n_1 n_2}^g}{\sum_{i,j} k_{ij}^g}$$

- 卷积的边值条件自选 (建议使用周期边值)
- 假设 \tilde{u} 是干净的原图, 待处理图片: $f = \mathcal{A}\tilde{u} + \sigma_1 \text{randn}$, 噪声级别 σ_1 参考值 $\frac{\max \tilde{u}}{100}, \frac{\max \tilde{u}}{50}$

2 作业要求

解压你提交的压缩包之后, 目录树应当如下 (下载示例):

```
hw1_学号_姓名/  
├── code/  
│   ├── README.txt  
│   ├── image/  
│   │   └── your test images  
│   └── else(subroutines, configure files, etc.)  
└── doc/  
    ├── report.pdf  
    ├── 理论作业.jpg[.png,](理论作业也可合并至 report.pdf 里)  
    └── else(*.tex, *.bib, *.docx etc.)
```

程序入口 程序应当至少支持以下 3 个参数

“输入图片路径”, “ λ ”, “输出图片路径”

将“输入图片路径”所指定的图片(自行设置 random seed 为此图片添加噪声)处理后输出到“输出图片路径”。可参考本文档最后的附录。

默认参数 请设置默认待处理图片及默认参数(包括优化参数, 迭代终止条件), 这组默认参数应当可以复现你在某个测试图片上某种噪声级别下的一个结果。

测试图片 hw1_学号_姓名/code/image, 至少包含本次作业指定测试图片(见评分)。

程序语言 请使用 matlab、python、C/C++ 之中的某一种完成上机作业。

上机报告 hw1_学号_姓名/doc/report.pdf 作为你的上机报告。

README 需要说明的 .

- 你的程序的用法. 在哪里设置参数(精确到具体文件的行数, 或说明如何通过命令行(如果支持)设置参数).
- 对于 matlab、python 用户, 程序中若使用了非 anaconda/matlab 自带模块/工具箱, 请在 code/README.txt 文件中说明.

- 对于 C/C++ 用户, 请提交源程序而非提交已经在你的系统中编译好的可执行文件, code/README.txt 中可说明其他人应该如何编译你的程序 (如果有多个源文件), 除标准库之外的其他依赖库 (如 opencv) 也请同时在 README.txt 中说明.

3 上机作业批改及评分

程序 60 分 去噪去模糊效果: PSNR,SSIM.

测试图片: peppers256.png,barbara.png; [此处为下载链接](#).

取模糊核方差 $\sigma = 1$, 对两个测试样例在两种噪声级别 ($\sigma_1 = 0.01, 0.02$) 下进行去噪去模糊 (不同噪声级别及图片可以用不同参数)

noise	index	peppers256	barbara
0.01	PSNR	24.86(+5.0)	25.02(+2.5)
	SSIM	0.935	0.83
0.02	PSNR	24.75(+3.0)	24.84(+2.0)
	SSIM	0.90	0.790

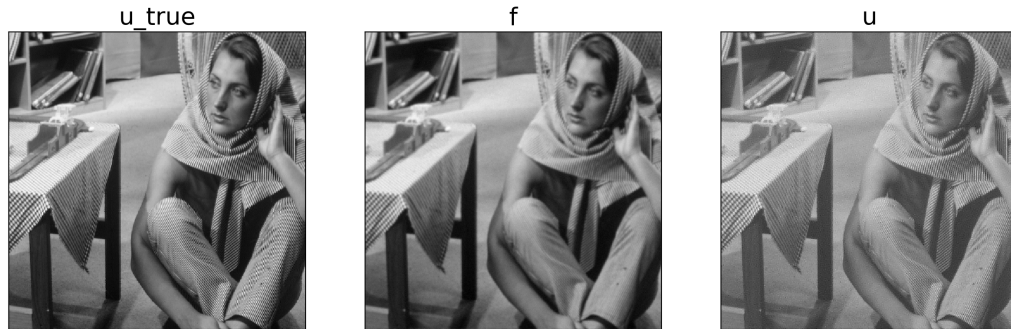
评分标准: 得分由 $\min_{\text{两种图片, 两种噪声}} \frac{\text{PSNR 提升量}}{\text{括号中的数字}}$ 决定.

$$\text{PSNR}(u) = 10 \log_{10}(\max(u)^2/\text{MSE}), \text{MSE} = \frac{1}{nm} \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} (u[i, j] - u_{\text{true}}[i, j])^2$$

SSIM 可参考 Matlab,Python(skimage) 提供的函数, 其中子窗口大小取为 15, 对各子窗口计算 SSIM 再求平均得到整张图片的 SSIM.

报告 40 分 **基本知识 (模型与算法的几个细节), 结果** 丰富性创新性, 排版. 你的结果应该在报告中用表格和图片两种方式呈现: PSNR 提升量 (表格), 原图 & 模糊带噪声图 & 算法恢复图 (图片). 例如

noise	index	peppers256			barbara		
		Origin	End	Max	Origin	End	Max
0.01	PSNR	24.861	30.430	30.451	25.016	27.732	27.822
	SSIM	0.9354	0.9554	0.9591	0.8319	0.8635	0.8831
0.02	PSNR	24.752	28.193	28.193	24.842	27.267	27.278
	SSIM	0.9025	0.9124	0.9267	0.7897	0.7591	0.7897

图 1: test image: barbara; noise level: $\sigma_1 = 0.01$

4 附录-程序入口及参数设置

4.1 matlab

matlab 可以利用 nargin 关键字来设置默认参数, 以下代码片段仅供参考:

```

% File Name: main.m
function u = main(image_path, lambda_weight, ...
    output_image, kernel_size, gaussian_sigma)
if nargin<1 image_path="image/fig1.png"; end % 输入图片名
if nargin<2 lambda_weight=1; end           % lambda
if nargin<3 output_image=...
    "image/fig1_result.png"; end         % 输出图片名
if nargin<4 kernel_size=15; end          % 卷积核大小
if nargin<5 gaussian_sigma=1; end       % 高斯核方差
% your code, e.g.
kernel = fspecial('gaussian', ...
    [kernel_size, kernel_size], gaussian_sigma)
f = imread(image_path)                   % 读取图片
u = mysolver(f, kernel, lambda_weight) % 自己实现这个函数
h = figure;
imshow(u)

```

```
print(h, output_image, '-dpng')    % 输出存储处理结果
end
```

按上述方式定义 main.m 文件, 那么我们可以登陆远程机器时在命令行做批量的测试. 其中 matlab 程序使用命令行参数用法

```
# current path: hw1_.../code/
matlab -nosplash -nodesktop -nodisplay -r \
    "main(image/fig1.png,1,15,1,image/fig1_result.png); exit"
```

4.2 python

以 python3 为例, 我们可以用 sys 模块读取命令行参数, 以下代码片段仅供参考

```
# File Name: main.py
import sys
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import mysubroutines # 自己实现子程序
argc = len(sys.argv)
# 从命令行接收参数
image_path = sys.argv[1] if argc>1 else "image/fig1.png"
lambda_weight = float(sys.argv[2]) if argc>2 else 1
output_image = (sys.argv[3] if argc>3
    else "image/fig1_result.png")
kernel_size = int(sys.argv[4]) if argc>4 else 15
gaussian_sigma = float(sys.argv[5]) if argc>5 else 1
# your code
kernel = mysubroutines.fspecial(
    [kernel_size, kernel_size], gaussian_sigma
)
f = mpimg.imread(image_path)
```

```
u = mysubroutines.mysolver(f, kernel, lambda_weight)
h = plt.figure()
ax = h.add_subplot(111)
ax.imshow(u)
h.savefig(output_image)
```

4.3 C++

C/C++ 可以在主函数中接收命令行参数, 以下代码片段仅供参考

```
/*File Name: main.cpp*/
#include <iostream>
/*something else*/

int main(int argc, char* argv[]) {
    /*variables declaration, definition, initialization*/
    if (argc < 2) {image_path="image/fig1.png";}
    else {image_path=argv[1];}
    if (argc < 3) {lambda_weight=1;}
    else {lambda_weight=std::atof(argv[2]);}
    if (argc < 4) {output_image="image/fig1_result.png";}
    else {output_image=argv[3];}
    if (argc < 5) {kernel_size=15;}
    else {kernel_size=std::atoi(argv[4]);}
    if (argc < 6) {gaussian_sigma=1;}
    else {gaussian_sigma=std::atof(argv[5]);}
    /*your code*/
}
```
