

homework1 说明

1 作业内容

1 (4 分) 参考 [课程第二讲 PPT](#) 第 77,78 页, 求 λ

2 (6 分) 用如下模型对图片进行去噪去模糊.

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

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,

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

- 卷积的边值条件自选, 请在提交的代码 `code/README.txt`¹ 中说明你能处理什么样边值的模糊算子. 可选边值请参考 matlab 中的 `imfilter` 函数或 python 函数 `scipy.ndimage.convolve`
- 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

$$\frac{\max \tilde{u}}{100}, \frac{\max \tilde{u}}{200}$$

¹后面会说明对目录树的要求

2 作业要求

解压你提交的压缩包之后, 目录树应当如下:

```
hw1_学号_姓名/  
├── code/  
│   ├── main routine(main.m, main.py, main.cpp, etc.)  
│   ├── 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 main routine 应当至少支持以下 5 个参数

“输入图片路径”, “ λ ”, “卷积核大小”, “高斯核方差”, “输出图片路径”

将“输入图片路径”所指定的图片处理后输出到“输出图片路径”. 这类程序入口是在服务器上运行程序的常用要求. 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.

测试图片 hw1_学号_姓名/code/image 下存放你自己所找的测试图片 (至少有一张), 图片文件名只允许包含英文字母和数字, 图片格式不限.

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 .

- 请说明你所做卷积 $\mathcal{A}u = k * u$ 所用的边值.
- 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 上机作业批改方式及评分点

批改方式:

1. 测试提交的作业中的默认图片及参数 & 助教任选图片加模糊加噪声对程序进行测试
2. 批改上机报告

评分参考项:

程序 去噪去模糊效果, 程序运行效率

报告 基本知识 (例如少数算法模型的关键细节), 结果丰富性创新性, 排版

4 附录-程序入口及参数设置

4.1 matlab

我们规定本次作业 main.m 文件中应定义 main 函数, 函数接收的前 5 个参数如前面程序入口中所述. 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, ...
    kernel_size, gaussian_sigma, result_image_path)

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 kernel_size=15; end % 卷积核大小
if nargin<4 gaussian_sigma=1.5; end % 高斯核方差
if nargin<5 result_image_path=...
    "image/fig1_result.png"; end % 输出图片名
% 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, result_image_path, '-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,1,15,1.5,image/fig1_result.png);_exit"
```

4.2 python

你的 python 程序应当支持命令行参数输入, 例如:

```
# current path: hw1_.../code/  
python ./main.py image/fig1.png 1 15 1.5 image/fig1_result.png
```

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  
kernel_size = int(sys.argv[3]) if argc>3 else 15  
gaussian_sigma = float(sys.argv[4]) if argc>4 else 1.5  
result_image_path = (sys.argv[5] if argc>5  
                     else "image/fig1_result.png")  
# 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(result_image_path)
```

4.3 C++

首先, 将 C++ 程序编译得到的可执行文件命名为 main, 这个可执行文件应当支持命令行参数, 例如:

```
# current path: hw1_.../code/  
./main image/fig1.png 1 15 1.5 image/fig1_result.png
```

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) { kernel_size = 15; }  
    else { kernel_size = std::atoi(argv[3]); }  
    if (argc < 5) { gaussian_sigma = 1.5; }  
    else { gaussian_sigma = std::atof(argv[4]); }  
    if (argc < 6) { result_image_path = "image/fig1_result.png"; }  
    else { result_image_path = argv[5]; }  
    /* your code */  
}
```
