DIP Homework Final - Deblur

江天源 1100016614

email: ws02298867@gmail.com

问题描述

在进行拍照的时候，经常会由于环境光照条件不好，为了拍入的照片有足够的亮度，需要使用更长的快门时间或者调高ISO感光度来进行拍摄。前者会在没有三脚架稳固时由于手抖而产生动态模糊（Motion blur）。为了恢复这些手抖模糊的照片而产生了一系列的去模糊（Deblur）算法。

本文实现了Yuan等人于2007年发布的论文 ***Image Deblurring with Blurred/Noisy Image Pairs*** 中所描述的deblur算法。

算法思路

Kernel Estimation

一个图片的模糊可以由以下公式表示：

$$B = I \otimes K$$

其中 B 是模糊的图片，I 是真实的图片（即没有模糊的图片），K 是模糊核（blur kernel）。这个公式表示，一个模糊的图片是可以表示成不模糊的图片卷积 K 得到。反过来说如果知道了 B 和 K，则可以推算出原图像 I。B 是已知的，因此deblur算法的主要任务就是估算 K 的值。

而 I 又可以表示为

$$I = N_D + \Delta I$$

，其中 N_D 为对在同样环境下，使用足够慢（慢到不会产生动态模糊）的快门和高ISO设置的条件拍到的图片 N 进行降噪（denoise）之后的结果。ΔI 则是两者的差值。一般而言 ΔI 非常小，$I \approx N_D$，可以使用 N_D 来估算 K 的值。

Yuen的论文中将公式中几个变量对应成了Matrix-Vector形式，将原问题转化为对方程

$$\mathbf{b} = \mathbf{A}\mathbf{k}$$

，即对l1-least squares问题的求解。论文中又对问题进行了Tikhonov正则化，将问题转化为优化问题：

$$\min_k \|\mathbf{A}\mathbf{k} - \mathbf{b}\|^2 + \lambda^2 \|\mathbf{k}\|^2, \text{subject to } k_i \geq 0, \text{and} \sum_i k_i = 1$$

，之后再使用Landweber方法对问题进行迭代求解：

> 1. 初始化，令 $\mathbf{k}^0 = \delta$，即迪拉克 δ 函数。
> 2. 更新，$\mathbf{k}^{n+1} = k^n + \beta(\mathbf{A}^T\mathbf{b} - (\mathbf{A}^T\mathbf{A} + \lambda^2\mathbf{I})\mathbf{k}^n)$
> 3. 令 $k_i^{n+1} = 0$ 若 $k_i^{n+1} < 0$，并令 $k_i^{n+1} / \sum_i k_i^{n+1}$，规范化其范围使其在 $[0, 1]$ 范围内。

> 4. (Hysteresis Thresholding) 在迭代的每一步中，使用 M_high 与 M_low 两个Mask，分别对当前迭代结束后的结果和下次迭代的结果进行thresholding，可以去掉kernle中的一些杂质信息，让结果更为精准。

令 $\beta = 1.0$ 迭代20到30次即可得到不错的结果。

Deconvolution

在算法的第一步中，估算得到了 K ，此时可以对 I 进行初步的估算。但是测试结果表现这样直接进行反卷积(deconvolution)的效果非常不好，有大量像波浪一样的奇怪花纹。这种现象被称为Gibs Phenomena，这种现象出现由于在图像中像边线、边界位置等地方或者因为一些算法而认为加进来的的不连续点存在，而且这些不连续点会在反卷积的过程中被加强，因而形成了图像上波纹。

Yuan对这个问题的处理方法是，利用公式 $I = N_D + \Delta I$ 其中 N_D 已经通过其他算法得到，只需利用公式

$$\Delta B = \Delta I \otimes K$$

和

$$\Delta B = B - N_D \otimes K$$

即

$$B - N_D = \Delta I \otimes K$$

算出 ΔI 的值，即可进一步求得 I 的值。而在反卷积求 ΔI 的过程中，由于 ΔI 和 ΔB 都很小，所以Gbis Phenomena产生的效果会很不明显。因而可以得到更为精确的 I 。

此外，论文中的反卷积使用了Richardson-Lucy（RL，也称Lucy-Richardson，LR）算法（又因为使用的是“残余”的图像 ΔI ，因此论文中称这种算法为残余RL算法（Residual RL））：

$$\Delta I_{n+1} = (K * \frac{\Delta B + 1}{(\Delta I_n + 1) \otimes K}) \cdot (\Delta I_n + 1) - 1$$

论文里面说的不是很清楚，在查阅相关资料后发现公式里的除法和点乘·都是基于元素的运算（element wise）；以及相关运算 $*$ 可以转换为卷积进行运算。得到公式：

$$\Delta I_{n+1} = (\frac{\Delta B + 1}{(\Delta I_n + 1) \otimes K} \otimes \hat{K}) \cdot (\Delta I_n + 1) - 1$$

其中

$$\hat{K}_{ij} = K_{(h-i)(w-j)}, 0 \le i, j \le h, w$$

这其实并不是RL算法本身的公式，区别在于 ΔB 和 ΔI 后的偏移量 $+1$ ——这是为了让 ΔI 的值在算法过程中非负。

这里我怀疑论文里有个小错误。论文中说的是让 ΔI 的值落在 $[0, 1]$ 区间，然而 ΔI 本身的值域是 $[-1, 1]$ ，因此 $\Delta I + 1$ 的取值范围应该是 $[0, 2]$ 。不过这一步的主要目的是让 ΔI 在算法过程中的取值范围合理，光结果而言这个做法是正确的。

De-ringing

虽然对 ΔI 进行反卷积可以很大程度上降低Gibs Phenomena导致的Ringing effect，但是Yuan觉得结果还是不够好。论文在RL的迭代过程中加入了一个增益控制（gain-control）系数 I_{gain} 。其值域在 $[0, 1]$ 之间，用来控制每次迭代的增益，从而进一步减小Ringing effect的出现。

此时RL算法的公式变为：

$$\Delta I_{n+1} = I_{gain} \cdot \{(\frac{\Delta B + 1}{(\Delta I_n + 1) \otimes K} \otimes \hat{K}) \cdot (\Delta I_n + 1) - 1\}$$

为了能够尽量保持平滑区域（不容易出现ring的区域），同时抑制边缘区域（容易出现ring的区域）的增益， 如下定义 I_{gain} ：

$$I_{gain} = (1 - \alpha) + \alpha \cdot \sum_l \| \nabla N_D^l \|$$

其中 α 为控制 I_{gain} 影响程度的系数，论文中 $\alpha = 0.2$ 。 ∇N_D^l 为降噪图 N_D 的高斯金字塔（Gaussian Pyramid，GP）的第 l 层的梯度。 $\sum_l \| \nabla N_D^l \|$ 则是在GP各层下，对应梯度的范数的总和。

这个公式我理解了很久。最开始我觉得 I_{gain} 是一个常数，这样的话 $\sum_l \| \nabla N_D^l \|$ 很容易就变得很大，从而使得 I_{gain} 的取值轻松超过了范围 $[0, 1]$ 。后来看了Yuan的论文中提到的另外一篇论文(***Gradient Domain High Dynamic Range Compression*** [Fattal et al. 2002])中对于这个公式的使用时提到的梯度的表示：

$$\nabla H_k = (\frac{H_k(x+1, y) - H_k(x-1, y)}{2^(k+1)}, \frac{H_k(x, y+1) - H_k(x, y-1)}{2^{k+1}})$$

此时才意识到 I_{gain} 也应该是对图中的某一像素点 (x, y) 而言的。因此得到更新的 I_{gain} 的定义为：

$$I_{gain}(x, y) = (1 - \alpha) + \alpha \cdot \sum_l \| \nabla N_D^l(x, y) \|$$

即每一个点有不同的增益系数 $I_{gain}(x, y)$ 。

利用这个增益控制的RL算法得到结果相对于纯Residual RL，在减少Ringing effect方面有着一定的提升。

Adding details

这是论文算法的最后一步，即向上一步通过增益控制RL得到的的图像中添加控制增益损失的细节。定义细节图层（detail layer）

$$I_d = I - \bar{I}, \bar{I}(x) = F(I(x))$$

其中 I 是使用纯ResidualRL得到的结果， $F(\cdot)$ 是一个低通过滤函数，因此细节图层 I_d 等效于高通过滤得到的结果。论文中 F 函数为联合双边过滤函数 [Eisemann and Durand 2004]：

$$F(I(x);Ig) = \frac{1}{Z_x} \sum_{x' \in W(x)} G_d(x - x') G_r(I(x) - I_g(x')) \cdot I_{x'}$$

其中 I_g 相当于Durand论文中的 I_p ，在算法中会考虑其权重来影响过滤的结果。

得到 I_d 之后将其加到前一步得到的 I_g 后即可得到补充了细节的最终结果。

算法实现

实现细节比较多，就不贴代码了。实现过程中的关键问题都在这里面包含了。

Kernel Estimation

总之我先随便找了一张图片作为 I ，然后用matlab的 `imfilter` 对图片加以动态模糊得到 B ，并使用 `imnoise` 函数生成同样亮度的无动态模糊但是有噪点图片作为 N ：

原图 `images/sculpture.jpg` ：

动态模糊后的图片 `images/blurred.jpg` ：

噪点图片 `images/noisy.jpg`

并尝试了两种算法进行降噪，一种是Yuan他们所用的wavlet based BLS-GSM（Bayesian Least Squares - Gaussian Scale Mixture）算法 [Portilla et al. 2003]，和Fast NLM（Fast Non Local Mean）算法 [Darbon et al. 2008]。经过测试后者速度较快，而且两者的降噪结果并无太大差别（Portilla的算法参数太难调了，而且论文提供的代码已经不适用于现在这个版本的matlab了，根据这个方法进行一些修改才能加入到项目里），因此就默认使用Darbon他们的算法。

降噪结果 `images/denoised.jpg`

之后的任务就是求解下述优化问题：

$$\min_k \|\mathbf{Ak}-\mathbf{b}\|^2+\lambda^2\|\mathbf{k}\|^2, k_i \geq 0, and \sum_i k_i = 1$$

在理解这个式子遇到了困难：论文里面并没有明确地解释图像和kernel的Matrix-Vector形式是什么意思。看了很多相关的资料也都没有找到具体的说明，最后终于在这个网站找到了解释。

简单来说就是论文中将 $B = I \otimes K$ 转化为 $\mathbf{b} = \mathbf{Ak}$ 形式是为了能够将卷积转换为矩阵-向量乘法，从而表示这个问题。因此需要预处理三者使他们进行乘法时能得到跟卷积一样的结果。

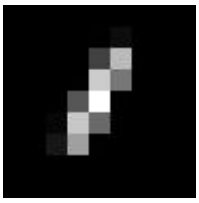
对于 B 和 K 只需将原矩阵 `reshape` 成 `[w*h 1]` 的矩阵即可。（其中 `w` 和 `h` 为两者原本的宽度和高度）

I 则比较复杂，需要将生成一个对应的大小为 `[w*h kw*kh]` 的新矩阵，其中 $\mathbf{A}(i+j*w,:)$ 为 $I(i,j)$ 与 K 进行卷积时所涉及的 I 中的元素。然而实际写的时候发现，在计算 A^TA 的时候非常非常的慢（毕竟是对两个大小为 `[w*h kw*kh]` 的矩阵做乘法）。

最后我找到了另一个去模糊算法的论文（***Robust Dual Motion Deblurring*** [Chen et al. 2008]）给出的代码，用了其中定义的一个类 `MatrixConvolve`，这个类重载了矩阵乘法运算符，使其在乘法时进行卷积，这样就免去了大量矩阵乘法的时间，而且适用于l1-ls优化问题。

在代码中实现了算法思路里面提到的方法，同时使用了Github上的一个解决l1-ls问题的开源项目（项目首页）。经过测试发现后者在速度和结果上都优于自己的实现，因此最后选择使用了后者（前者的代码依然保留）。

算法估算出来的blur kernel `images/kernel_estimated.jpg`：



真实的blur kernel `images/kernel_true.jpg`：

对比两者中心位置，可以发现算法对kernel的估计还是非常准确的。

Deconvolution

先直接根据公式 $B = I \otimes K$ 用 B 和 K 进行反卷积（deconvolution）得到一个初步估算的 I 。分别测试了两种matlab自带的deconv方法：

1. `deconvreg`，瑕疵非常的显眼 `images/deblurred_reg.jpg`：

2. `deconvlucy`，结果已经很不错了，不过图像边界，以及图像中间的一些边缘位置（比如雕像的袖子周围）的ringing effect还是挺明显的 `images/deblurred_lucy.jpg`：

这两个自带算法的结果并不是很理想，所以我继续尝试论文中的算法 Residual RL。

首先考虑到后面的增益控制RL的需要，我重新实现了一个RL deconvolution函数`deconvglucy`。函数可以根据参数 α 的值来决定增益系数的大小。令 $\alpha = 0.0$ 则 $I_{gain} = 0$，则此时该算法理论上与之前的RL算法是相同的。

理应……是这样的……然而实现了Residual RL后却得到了这样的结果`images/deblurred_residual_RL_bad.jpg`：

可以看出边缘有非常明显的白线。查看代码之后才发现，算法过程中的卷积使用的命令是（例如）：

```
I = conv2(B, K, 'same');
```

这样虽然卷积出来的结果跟之前是一样的，但是实际上只是强行返回了原大小的矩阵，边缘部分是有瑕疵的。因此我（顺便用FFT）重新实现了一个卷积函数 `fftconv2`，在其中考虑了边界部分的处理：

```
1. 将原矩阵向四周扩大并用`1`填充
2. 卷积
3. 返回与原来同等大小的相应部分的矩阵
```

测试之后得到了正常的结果 `images/deblurred_residual_RL.jpg`：

De-ringing

上一步得到的结果已经很不错了，不过仔细看的话图像边界和图中边缘区域仍有以下波纹。因此不满足于现状继续实现论文后面的算法——Gain-controlled RL。

由于前一步中已经实现了支持增益系数的新的RL函数，这一步主要的任务是完成其中增益系数值的计算。虽然 I_{gain} 本身不是一个固定的值，但是对于给定像素坐标 (x, y) 而言，I_{gain} 是唯一确定的。因此可以在函数的预处理阶段直接完成整个 I_{gain} map的计算。

这部分的实现中，GP的计算实现借用了Matlab File Excange中的代码 `Gscale`。再对于GP中的每一层，算好其坐标梯度 `imgradientxy`（梯度计算使用的中心差分法 `CentralDifference` [Fattal et al. 2002]）。遍历整个图像，计算其中每一个点的梯度的2-范数（其实相当于 $\sqrt{x^2 + y^2}$ ），再将该点对应GP中的所有层进行求和即可。

最后在 `deconvglucy` 除了最后一次迭代以外的结果点乘上 I_{gain} 即可。

结果 `images/deblurred_gain_controlled_RL.jpg`：

差别和Residual RL之间不那么明显，不过观察图片边界和图中边缘位置还是可以看出来Ringing effect进一步减轻了不少。

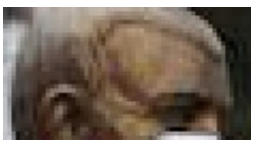
Adding Details

终于到了最后一步！这一步比较简单，在论文里其实都没有专门列出一个章节来说明。我直接从Matlab File Exchange找到了一个Joint/Cross Bilateral Filtering算法的实现代码，并在我的项目中引用了它。

使用这个算法就能得到 $\bar{I}$，从而进一步得到细节图层I_d。将其加到$Gain-controlledRL$的结果I上后得到最终结果。

图像的细节比起之前有所提升。例如在雕像额头处：

`Gain-controlled RL`：



`Gain-controlled RL with details`

很明显可以看出后者比起前者在太阳穴附近以及头发上的纹理有了细节上的提升。

最终结果 `images/deblurred_detail_RL.jpg`：

对比原图 `images/sculpture.jpg`：

与原图已经几乎相差无几了。至此实现完毕。

算法测试

- 同一个kernel[5x5]测试另一个图片：

  `cameraman.tif`

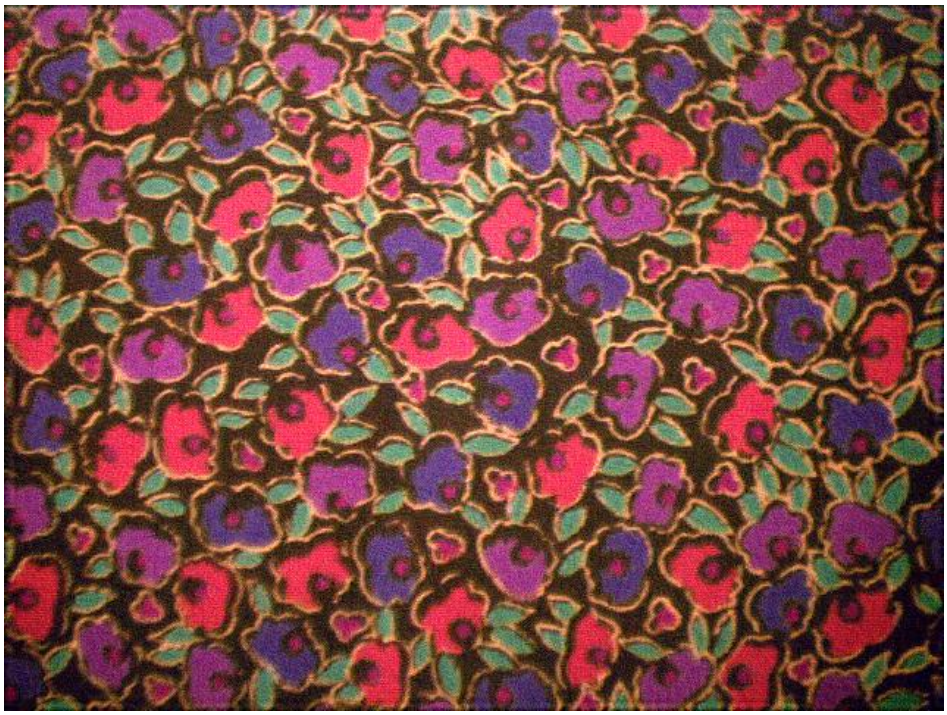
  

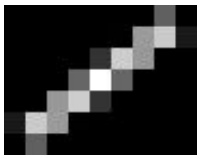
  结果不错。

- 更大的kernel[9x9]测试更大的图片

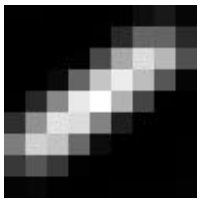
  `fabric.png`



`kernel_true`



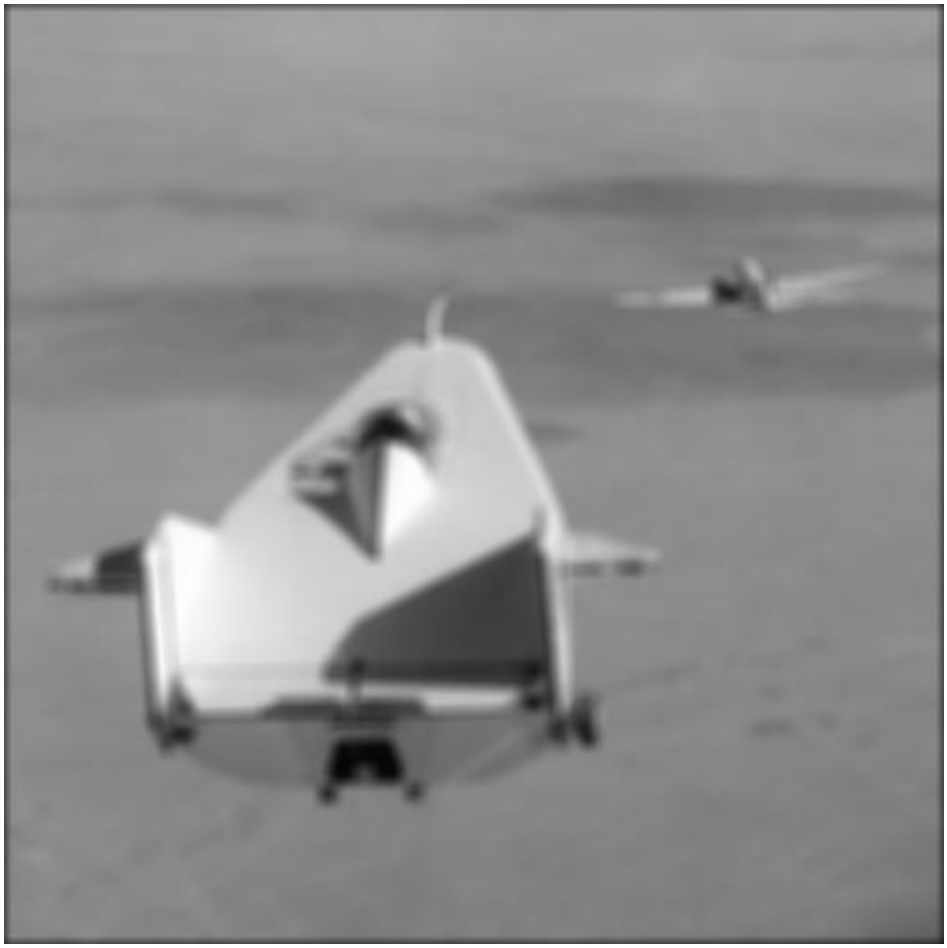
`kernel_estimated`

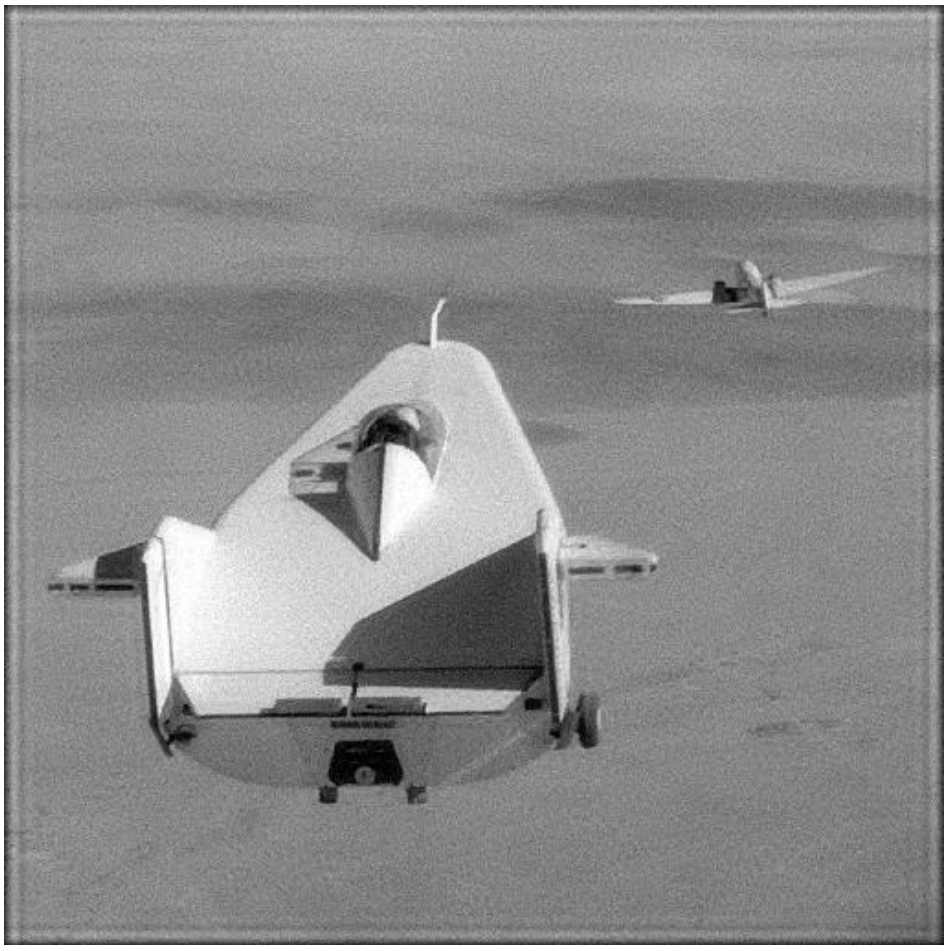


结果也不错。不过kernel上的估算稍有偏差，虽然大体走向是对的，但是分布上有些向外扩散的趋势。结果图片放大之后仔细观察也发现还有一些噪点，应该是由于降噪阶段的结果不够理想。

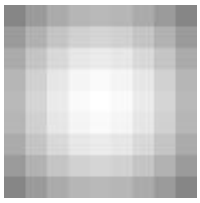
- 高斯模糊kernel测试

  `liftingbody.png`

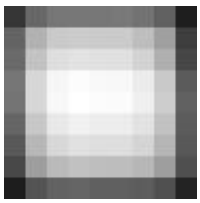




`kernel_true`



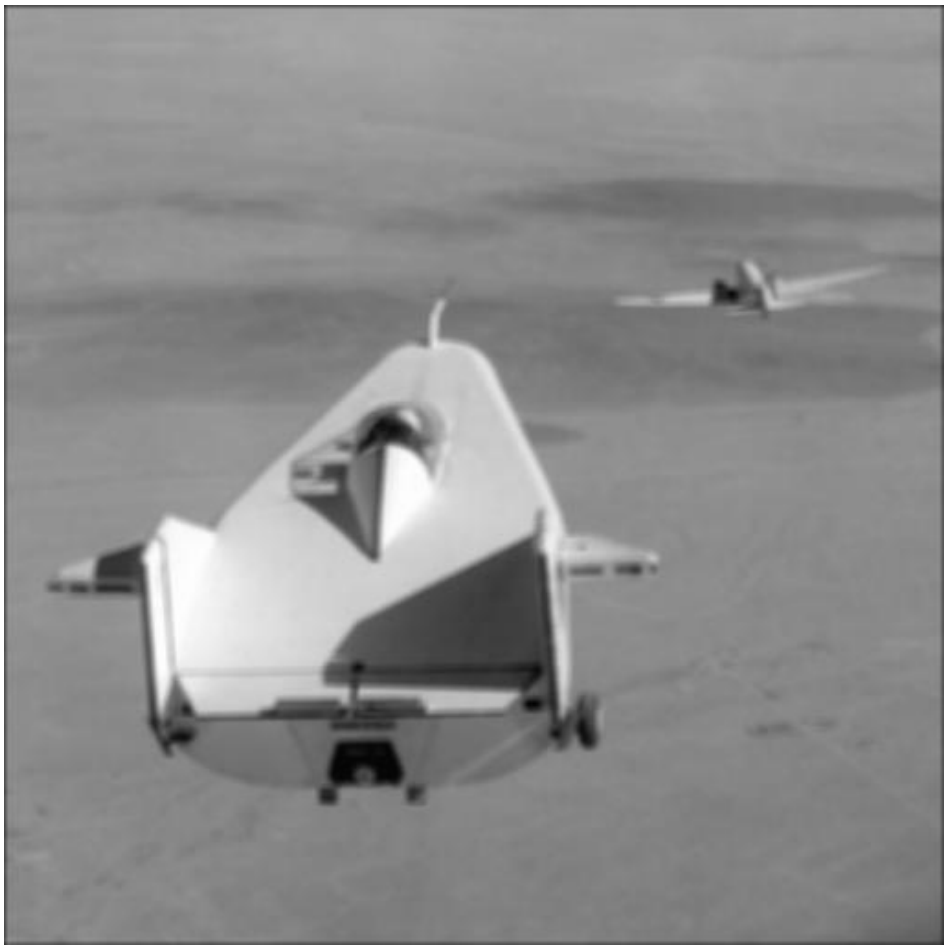
`kernel_estimated`

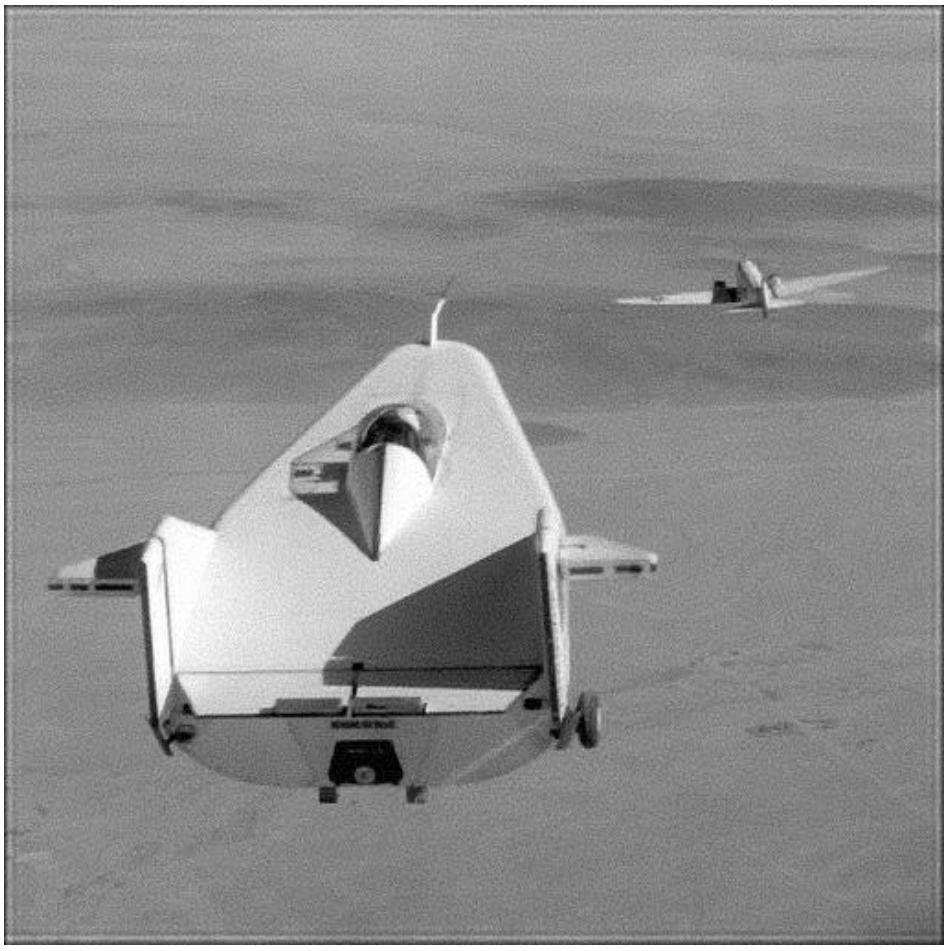


这种模糊一般出现拍照时未对好焦的情况。虽然论文是针对“motion blur”进行去模糊的算法，可以看出对于高斯模糊也有不错的deblur结果。不过去模糊后图中有一些明显的噪点，对比降噪图 N_D 发现，在降噪这一步结果就不够理想。由此可以知道该算法会很大程度上受限于降噪的结果好坏。

- 随机kernel测试

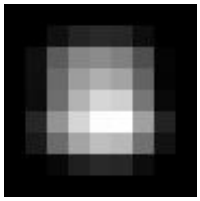
`liftingbody.png`





`kernel_true`

`kernel_estimated`



随机情况下去模糊的结果意外的不错，不过从kernel来看虽然大体范围上比较准确，但是kernel中一些不均匀地地方却没有准确地估算。

- 随机kernel测试 2

`onion.png`

`kernel_true`

`kernel_estimated`

对于这张图去模糊的效果就不那么好了，图中可以看到明显的Ringing effect，而且本身清晰度也不够。从kernel来看估算也很不准确。看来这个算法对于完全随机的kernel估算并不总能得到令人满意的结果。

总结与感想

Yuan等人的论文所描述的迭代算法包括以下四个步骤：

1. 估算kernel K
2. 计算Residual RL算法反卷积还原的图像 I_r
3. 计算Gain-controlled RL反卷积还原的图像 I_g
4. 计算细节图层 I_d ，并将其叠加到 I_g 上

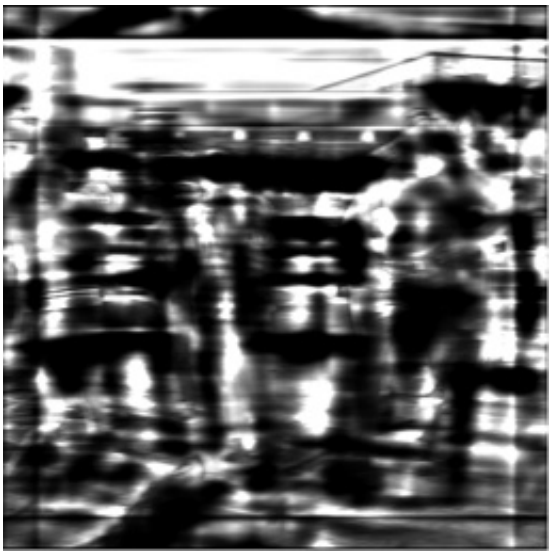
反复执行这几步直到得到足够好的结果为止。

这个算法已经有很不错的表现了，但是还有以下几点局限：

1. 要求降噪算法有较好的表现，因为论文利用 $I = N_D + \Delta I$ 时 N_D 会很大程度影响总体的结果。
2. 对于较大图片或者较大的kernel，在求解l1-ls问题估算kernel时，很难在合理时间内得到足够精确的结果。
3. 算法对于人工合成的模糊估算的kernel相当准确，但是在现实中可能产生的复杂情况（或者随机生成的kernel），表现并不令人满意。

这些问题还有待该领域的进一步研究解决。

数字图像处理这门课一学期下来真是学到了各方各面的很多知识。虽然课上的内容有很多的公式，但听了老师的讲解而且下来认真思考都还是能掌握。平时的作业都比较简单，不过在理解实现的那些小功能以及阅读matlab自己的相关实现时感觉获益匪浅。期末这次大作业更是如此，从一篇论文出发，边实现边进行相关的调研，直到完全实现算法。这个过程对于涉及到的每一个公式、每一种算法都去进行了研究，虽然遇到了很多的困难，但是最后（应该）都解决了。看着算法一步一步改进，从最开始的（开始两张其实用的原始图片跟后来不一样了，所以尺寸不一样）：



和

从这种完全可以当成之前stylize的作业提交的输出_(´ཀ`」 ∠)_，到最后的：

逐步地变得能跑出预想的结果，还是非常有成就感的。

参考文献

1. Yuan, Lu, et al. “Image deblurring with blurred/noisy image pairs.” *ACM Transactions on Graphics (TOG)* 26.3 (2007): 1.

2. Rouf, Mushfiqur. “A study on blur kernel estimation from blurred and noisy image pairs.” *CPSC548 UBC* 7 (2008).

3. Fattal, Raanan, Dani Lischinski, and Michael Werman. “Gradient domain high dynamic range compression.” *ACM Transactions on Graphics (TOG)*. Vol. 21. No. 3. ACM, 2002.

4. Eisemann, Elmar, and Frédo Durand. “Flash photography enhancement via intrinsic relighting.” *ACM transactions on graphics (TOG)*. Vol. 23. No. 3. ACM, 2004.

5. Portilla, Javier, et al. “Image denoising using scale mixtures of Gaussians in the wavelet domain.” *Image Processing, IEEE Transactions on* 12.11 (2003): 1338–1351.

6. Darbon, Jérôme, et al. “Fast nonlocal filtering applied to electron cryomicroscopy.” *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium* on. IEEE, 2008.

7. Chen, Jia, et al. “Robust dual motion deblurring.” *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008.

8. David Jacobs. “Correlation and Convolution”. *Class Notes for CMSC 426*, Fall 2005.

9. Denoising Fluorescence Images

10. Mushfiqur’s Projects :: Blur Kernel Estimation from Blurred and Noisy Image Pairs.

11. Richardson–Lucy deconvolution - Wikipedia, the free encyclopedia.

12. Simple Matlab Solver for l1-regularized Least Squares Problems.