National Tsing Hua University
Department of Electrical Engineering
EE6620 Computational Photography, Spring 2021

Homework #2
Non-Blind Deblurring

Assigned on April 14, 2021

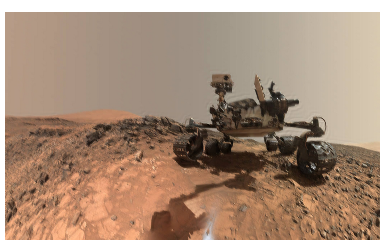
Due by May 5, 2021

Homework Description

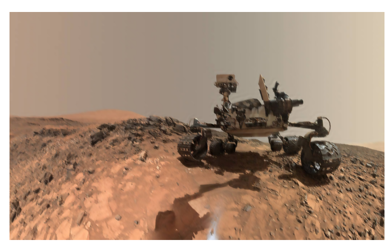
A slow shutter speed will introduce blurred images due to camera shake. The objective of this assignment is to implement several non-blind deblurring algorithms and analyze the effect on blurred images. In part 1 and 2, you will implement Richarson-Lucy (RL) deconvolution [1] [2] and its bilateral variant (BRL) [3]. And in part 3, you are going to solve deblurring problem in total variation model.

There are two types of image sets in this assignment, synthetic image and real image. The **Curiosity-small** and **Curiosity-medium** blurred images were synthesized using small and medium blur kernel sizes respectively. For simplicity, the blurring was performed in non-linear $(R'/G'/B')$ domain. The **CVBook** image was taken by a camera phone, and its kernel was estimated roughly using prior knowledge.

Blurred



RL deblur



BRL deblur

Table 1: Scores

Items		Score
Implementation (55%)	RL deconvolution	20%
	BRL deconvolution	25%
	Total variation regularization	10%
Experiment& Free Study (45%)	Experiments	25%
	Free Study	20%

1 Implementation (55%)

The reference answers of RL, BRL and total variation regularization are provided in the homework file (HW2/ref_ans). You can compare the PSNR between your results and reference answers by the function `PSNR_UCHAR3()` in load_psnr.py.

There are some details you should comply with in implementation:

1. Perform RL/BRL for the three color channels separately.
2. Use **floating-point** operations and normalize the intensity I to **[0, 1]** before deblurring.
3. Normalize the blur kernel K such that $\sum k_i = 1$, before using it.
4. For doing convolution and padding, the **symmetric boarder condition** (Appendix I) is used in reference answer. However, you can try some different boundary conditions in your report.
5. RL/BRL can work for different choices of initial images $I^{(0)}$. The blurred image (original image) is a typical choice, but you can try other initial images (for example, a flat black image) in your report.
6. Gamma factor = 2.2

1.1 RL deconvolution (20%)

RL deconvolution can be express as

$$I^{t+1} = I^t \circ [K^* \otimes \frac{B}{I^t \otimes K}], \tag{1}$$

where

$$K^*(i,j) = K(-i,-j). \tag{2}$$

Energy function of RL can be formulated as

$$E(I) = \sum((I \otimes K) - B \circ \ln(I \otimes K)). \tag{3}$$

Finish the function `RL()` and `energy_RL()` in deconv_RL_BRL.py. `RL()` function should be able to process the input image by RL deconvolution and `energy_RL()` function should be able to calculate the energy of the deblurred results of RL. The input/output argument of the function should not be changed. The PSNR should be larger than 60 dB for `RL()` and the error for `energy_RL()` should be smaller than 0.05% .

Please generate the following 2 images in /result for TA to examine your work.

a. RL on curiosity_small, with iteration=25.
   (compared with ref_ans/curiosity_small/rl_deblur25.png)
b. RL on curiosity_medium, with iteration=55.
   (compared with ref_ans/curiosity_medium/rl_deblur55.png)
c. Calculate energy of RL on results above.

1.2 BRL deconvolution (25%)

BRL deconvolution can be express as

$$I^{t+1} = \frac{I^t}{1 + \lambda \cdot \nabla E_B(I^t)} \circ (K^* \otimes \frac{B}{I^t \otimes K}), \tag{4}$$

where

$$K^*(i, j) = K(-i, -j), \tag{5}$$

and

$$\nabla E_B(I^t) = 2 \cdot \sum_{y \in \Omega} (\exp(-\frac{|x-y|^2}{2\sigma_s}) \cdot \exp(-\frac{|I(x) - I(y)|^2}{2\sigma_r}) \cdot \frac{I(x) - I(y)}{\sigma_r}). \tag{6}$$

In (6), x is the center pixel and y are nearby pixels in Ω.

Energy function of BRL can be formulated as

$$E(I) = \sum \left\{ \sum \{(I \otimes K) - B \circ \ln(I \otimes K)\} + \lambda E_B(I) \right\}, \tag{7}$$

where

$$E_B(I) = \sum_x \sum_{y \in \Omega} \exp(-\frac{|x-y|^2}{2\sigma_s})(1 - \exp(-\frac{|I(x) - I(y)|^2}{2\sigma_r})) \tag{8}$$

Finish the function `BRL()` and `energy_BRL()` in deconv_RL_BRL.py. `BRL()` function should be able to process the input image by BRL deconvolution and `energy_BRL()` function should be able to calculate the energy of the deblurred results of BRL. The input/output argument of the function should not be changed. The PSNR should be larger than 55 dB for `BRL()` and the error for `energy_BRL()` should be smaller than 0.05% .

Please generate the following 4 images in /result for TA to examine your work.

a. BRL on curiosity_small, with
   $\lambda = 0.03/255$, iteration=25, $\sigma_r = 50/255^2$, $r_K = 6$, $r_\Omega = 0.5 r_K$, $\sigma_s = (r_\Omega/3)^2$.
   (compared with ref_ans/curiosity_small/brl_deblur_lam0.03.png)
b. BRL on curiosity_small, with
   $\lambda = 0.06/255$, iteration=25, $\sigma_r = 50/255^2$, $r_K = 6$, $r_\Omega = 0.5 r_K$, $\sigma_s = (r_\Omega/3)^2$.
   (compared with ref_ans/curiosity_small/brl_deblur_lam0.06.png)
c. BRL on curiosity_medium, with
   $\lambda = 0.001/255$, iteration=55, $\sigma_r = 25/255^2$, $r_K = 12$, $r_\Omega = 0.5 r_K$, $\sigma_s = (r_\Omega/3)^2$.
   (compared with ref_ans/curiosity_medium/brl_deblur_lam0.001.png)

d. BRL on curiosity_medium, with
   $\lambda = 0.006/255$, iteration=55, $\sigma_r = 25/255^2$, $r_K = 12$, $r_\Omega = 0.5r_K$, $\sigma_s = (r_\Omega/3)^2$.
   (compared with ref_ans/curiosity_medium/brl_deblur_lam0.006.png)
e. Calculate energy of BRL on results above.

1.3 Solving deblurring problem by total variation regularization (10%)

You are going to implement the Total Variation algorithm using ProxImaL[4]. A example usage in `TVL1.py` shows how to implement Total Variation model in norm 1 using ProxImaL. In TVL2.py, you need to change the data term from norm 1 to norm 2. In TVpoisson.py, you need to change the data term from norm 1 to model the noise in the blurring process as Poisson distribution (hint: use ProxImaL function `poisson_norm()`). Your TVL2 and TVpoisson result should be similar with ref_ans/curiosity_medium/deblur_edgetaper_norm2.png and ref_ans/curiosity_medium/deblur_edgetaper_poisson.png (PSNR > 60 dB).

2 Experiment & Free Study (45%)

There are two parts of problems: Experiments and Free Study. Finish problems in experiments then choose at least four interesting topics of deblurring for your free study. You should answer the problems by following the two-step research flow: **Assumption** and **Justification**. Based on your observation, repeatedly propose an assumption, and justify your idea by experiments. After you reach a satisfying result, clearly describe your ideas and justification in the report. Grading criteria for this part are based on the clarity of the report and rigorousness of the justification. Note that you can discuss as more topic as you want in free study; however, we will consider both breadth and depth in grading.

2.1 Experiments (25%)

a. (10%) A conceptual figure (Figure 1) shows how to evaluate your deblurred result by 1-D DFT scanline. Try to plot a same figure for the deblurred results and a non-blurred image data/curiosity.png. How do you plot the figure? What is the meaning of the figure? How this figure can help you to explain the result? Do you apply any additional steps? Why?
b. (15%) Take two blurred images (one with straight kernel and one with curl kernel), estimate their blur kernel by yourself, and generate your own deblurred images using BRL deconvolution. A reference method to photograph a blur image with blur kernel is shown in (Figure 2). Document the deblur flow you choose for your images. For example, how you choose your parameters, some extra preprocess on your blur kernels. Also compare the the deblur effect for the two kernels.

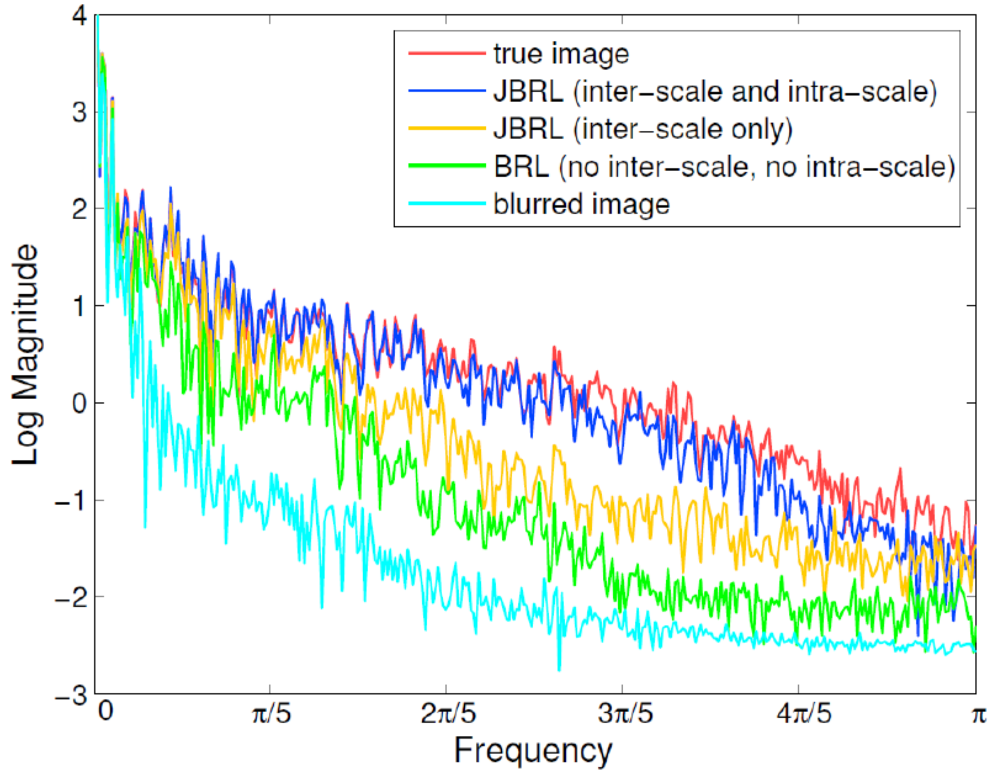


Figure 1: DFT curves of a 1-D scanline

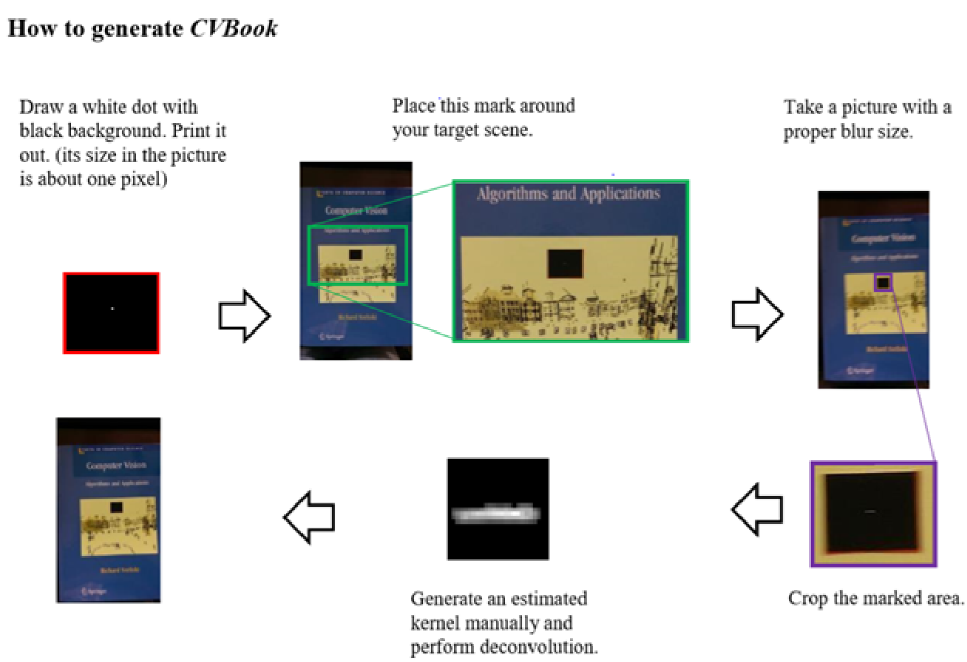


Figure 2: How to generate CVBook

2.2 Free study (20%)

Please find at least four interesting aspects of the deblurring image flow as your research topics. You can also choose your research topics from the following problems.

1. Compare the RL and BRL results of Curiosity-small and Curiosity-medium. Then compare the BRL results of Curiosity-medium with different parameters. There are many parameters you can vary like λ, σ_r, iteration counts. Also, you could try different boarder condition of convolution or different r_Ω to r_K ratio. Please change at least 2 parameters and compare the difference.
2. Compare the BRL results of Curiosity-medium with different boarder condition.
3. Compare some different λ on TVL1 deconvolution.
4. Compare the results of TVL1, TVL2 and TVpoisson.
5. Try to speed up BRL and report the execution time difference.
6. Use energy function to estimate the convergence of RL and BRL.

3 Deliverable

- HW2_studentID
  - code
    - deconv_RL_BRL.py
    - TVL2.py
    - TVpoisson.py
  - my_RL_BRL_result
    - RL_s_iter25.png
    - RL_m_iter55.png
    - BRL_s_iter25_rk6_si50.00_lam0.03.png
    - BRL_s_iter25_rk6_si50.00_lam0.06.png
    - BRL_m_iter55_rk12_si25.00_lam0.001.png
    - BRL_m_iter55_rk12_si25.00_lam0.006.png
    - deblur_edgetaper_norm1.png
    - deblur_edgetaper_norm2.png
    - deblur_edgetaper_poisson.png
  - my_deblur
    - my_image.png
    - my_kernel.png
    - my_deblur.png
  - HW2_report_studentID.pdf

Compress your whole folder HW2_studentID in HW2_studentID.zip and submit to iLMS.

Note that wrong file delivery or arrangement will get 5% punishment.

References

[1] W. H. Richardson, "Bayesian-based iterative method of image restoration∗," *J. Opt. Soc. Am.*, vol. 62, pp. 55–59, Jan 1972.

[2] L. B. Lucy, "An iterative technique for the rectification of observed distributions," *Astron. J.*, vol. 79, pp. 745–754, 1974.

[3] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum, "Progressive inter-scale and intra-scale non-blind image deconvolution," *ACM Trans. Graph.*, vol. 27, p. 1–10, Aug. 2008.

[4] F. Heide, S. Diamond, M. Nießner, J. Ragan-Kelley, W. Heidrich, and G. Wetzstein, "Proximal: Efficient image optimization using proximal algorithms," *ACM Trans. Graph.*, vol. 35, July 2016.

4 Appendix

4.1 Appendix I

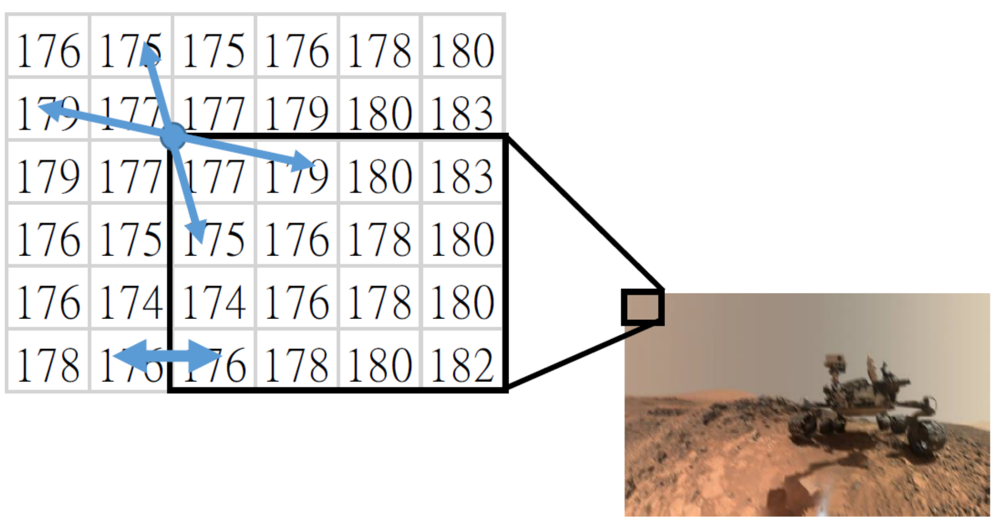


Figure 3: symmetric boarder condition