Image Processing Homework 3

L091084 邱楷涵

1 Method

In this homework, our purpose is to blind deblur the image. The main problem is how to determine the blur kernel. Suppose the model is defined as

$$y = x * k + n,$$

where y is the blurred image which a clear image convoluted with a blur kernel k and plus noise n. Our purpose is to fit the clear image $\hat{x}$ and the filter $\hat{k}$.

At here, we use the method (Jiangxin Dong and Jinshan Pan (2017) Blind Image Deblurring with Outlier Handling) to find the suitable kernel. This method is to find the minima estimator of the model

$$\arg\min_{x,k} R(x * k - y) + \gamma P_k(k) + \lambda P_x(x),$$

where $P_x(x), P_k(k)$ are priors penalty; γ and λ are weights and R is the robust function.

It is clearly that the estimators have no closed form; hence, they chose EM algorithm for the optimizer which defined as:

$$\hat{x}^{(t+1)} = \arg\min_x R(x * k^{(t)}) + \lambda P_x(x), \tag{1}$$

$$\hat{k}^{(t+1)} = \arg\min_x R(x^{(t)} * k) + \lambda P_k(k). \tag{2}$$

To maintain the sparsity of kernel, they chose the lasso penalty to regularize k, namely, $P_k(k) = \|k\|_1$. For the image penalty, they use hyper-Laplacian prior and set $P_x(x) = \|\nabla x\|_{0.8}$. Finally, the robust function is defined as

$$R(z) = \frac{z^2}{2} - \frac{\log(a + e^{bz^2})}{2b}$$

However, due to the choice of the penalty, the algorithm still has no closed form, it still reduce some calculating time. To solve the minimum of the function (1), they use IRLS method. At each iteration, it is needed to solve

$$x^{(t+1)} = \arg\min_x \sum_p [w^x|(x * k - y)_p|^2 + \lambda(w_h^x|(\partial_h x)_p|^2 + w_v^x|(\partial_v x)_p|^2)],$$

where $w_h^x = |(\partial_h x^{[t]})_p|^{-1.2}$,$w_v^x = |(\partial_v x^{[t]})_p|^{-1.2}$, $w^x = \frac{R'((x^{(t)}*k^{(t)}-y)_p)}{(x^{(t)}*k^{(t)}-y)_p)}$ and $R'()$ is the derivative function of R which has formula

$$R'(z) = z - \frac{ze^{bz^2}}{(a + e^{bz^2})}.$$

Finally, the minimum problem is a weighted least squares problem and can solved by the conjugate gradient method.

Similarly, (2) can be solved by

$$k^{(t+1)} = \arg\min_k \sum_p [w_h^k|(\partial_h x^{(t+1)} * k^{(t)} - \partial_h y)_p|^2 + w_v^k|(\partial_v x^{(t+1)} * k^{(t)} - \partial_v y)_p| + \gamma w^k |k_p^{(t)}|^2],$$

where $w_h^k = \frac{R'((\partial_h x^{(t+1)}*k^{(t)}-\partial_h y)_p)}{(\partial_h x^{(t+1)}*k^{(t)}-\partial_h y)_p)}$,$w_v^k = \frac{R'((\partial_v x^{(t+1)}*k^{(t)}-\partial_v y)_p)}{(\partial_v x^{(t)}*k^{(t)}-\partial_v y)_p)}$ and $w^k = \frac{1}{|k_p^{(t)}|}$.

According to the article, they convert the color image to grayscale to fit the kernel $\hat{k}$. After that, using the fitting filter $\hat{k}$ to initialize the kernel parameter in the non-blind model, here we combine the blind model(S. Cho, J. Wang (2011)Handling outliers in non-blind image deconvolution) and get the final result. This method is modified from "High-Quality Motion Deblurring from a Single Image" which mentioed in the class. Different from it, they add another prior, m, which is a binary variable with value 0, if the pixel is outlier, and value 1 otherwise.

Similarly to previous method, both of them want to find the MAP for

$$x_{MAP} = \arg\max_x P(x|y,k),$$

where L is a likelihood function. According to the Bayes' theorem, the equation can be rewritten as

$$\begin{aligned} x_{MAP} &= \arg\max_x P(x|y,k) \\ &= \arg\max_x P(y|x,k)P(x) \\ &= \arg\max_x \sum_{m \in M} P(y,m|x,k)p(x) \\ &= \arg\max_x \sum_{m \in M} P(y,|m,x,k)P(m|y,k)P(x), \end{aligned} \tag{3}$$

where M is the space of m. They define the latent prior as

$$P(x) = \frac{e^{-\lambda\phi(x)}}{Z},$$

where Z is normalization constant and $\phi(x) = \sum_p\{|(\partial_h x)_p|^\alpha + |(\partial_v x)_p|^\alpha\}$. Here choose $\alpha = 0.8$. Next, since the noise are assumed independent, the likelihood for each pixel is set as

$$P(y_p|m,x,k) = \begin{cases} N(y_p|f_p,\sigma) & ,m_x = 1 \\ C & ,m_x = 0 \end{cases}$$

where $f = x * k$, N is normal distribution with standard deviation σ.

For the prior $P(m|x,k)$, they suppose

$$p(m_p = 1|f_p) = \begin{cases} P_{in} & ,f_p \in \text{DR} \\ 0 & ,\text{otherwise} \end{cases}$$

where DR is dynamic range, and P_{in} is the probability for the pixel is inlier.

Since solving the equation is hard, them decide use EM algorithm to solve the parameter.

For the E step, define

$$\begin{aligned} Q(x,x^{(t)}) &= E(\log P(y,m|x,k)) \\ &= E(\log P(y|m,x,k) + \log P(m|x,k)) \\ &= -\sum_p \frac{E(m_x)}{2\sigma^2}|y_p - f_p|^2, \end{aligned}$$

where

$$\begin{aligned} E(m_p) &= P(m_X = 1|y,k,x^{(t)}) \\ &= \begin{cases} \frac{N(y_x|f_p^{(t)},\sigma)P_{in}}{N(y_x|f_p^{(t)},\sigma)P_{in}+C(1-P_{in})} & ,f_p \in \text{DR} \\ 0 & ,\text{otherwise} \end{cases} \end{aligned}$$

and $f^{(t)} = x^{(t)} * k$.

For the M step the updated latent image x is defined as

$$x^{(t+1)} = \arg\max_x \{Q(x, x^{(t)}) + \log P(x)\}$$

which is equivalent to minimizing

$$\sum_p w_p^m |(y - x * k)_p|^2 + \lambda \phi(x), \tag{4}$$

where $w_p^m = \frac{E(m_p)}{2\sigma^2}$. To approximate ϕ, they use IRLS method and rewrite the equation (4) to

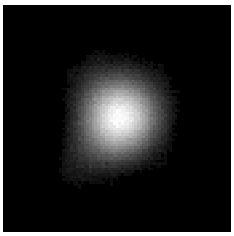
$$\sum_p w_p^m |(y - x * k)_p|^2 + \lambda(w_h^x |(\partial_h x)_p|^2 + w_v^x |(\partial_v x)_p|^2)]$$

and the equation can be solved by conjugate gradient method as well. Hence, we can get the latent image. Eventually, compute the PSNR value by tensorflow.image.psnr.

2 Results

2.1 input1

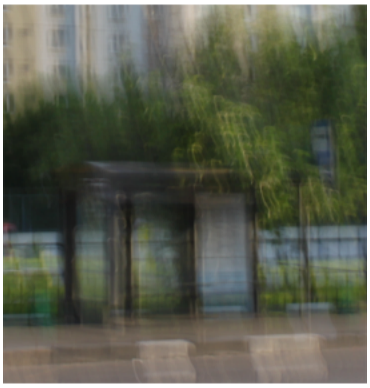
The problem of image is Gaussian blur, it is difficult to determine buildings edge. Here, we first estimate the kernel by the Jiangxin Dong and Jinshan Pan(2017) method.

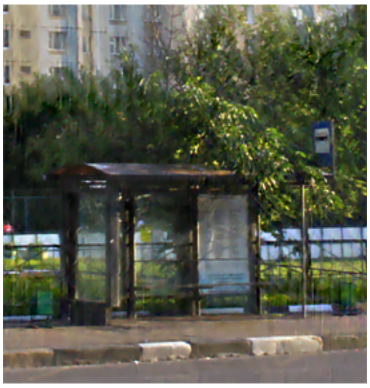


The kernel indeed looks like the Gaussian kernel but has some small pattern on bottom left. Using this kernel to the initial kernel for Cho's method, we get the restored image

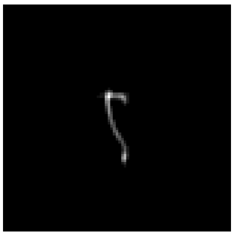
Although the image is still blur, the edge of the buildings is clearer than input. Otherwise, it still has some ring effect but not significant. In particular, if we observe carefully, near by 101 is non-natural. Eventually, the PSNR value from 21.6 goes to 22.6.

2.2 input2

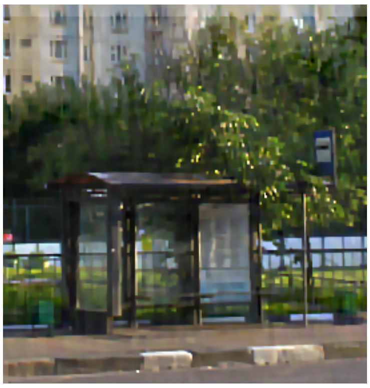




The problem of image is motion blur, the main problem is to determine the motion path. Similarly, we first estimate the kernel.



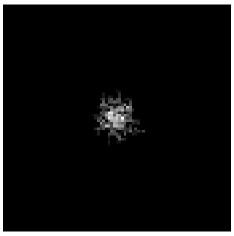
The kernel indeed looks like the motion kernel and the main direction is from top left to bottom right. Similarly, use this initial kernel and get the restored image



The recovery looks great but has some panning from the original picture. It just looks like use different angle of shot. However, the PSNR value from 19.6 goes to 16.2 even though I cannot distinguish which is the original image. In my opinion, compute PSNR for blind deconvolution is not a good standard since a shot angle make big square error.

2.3 input3

The problem of image is Gaussian blur adding the noise, the big problem is the probability of the noise is too high so that most of general method like adaptive medium filter is useless. Hence, we directly fit the kernel from the input.



It cannot certain what kernel is it since there are many sparse point in the middle. Next, we use this kernel for initial kernel as well.

The restoration has a lot of noise at top middle and some noise at other where. However, license plate clearer than the input. Finally, the PSNR value from 19.6 goes to 22.3.

References

[1] J. Dong and J. Pan. Blind Image Deblurring with Outlier Handling. In ICCV, 22-29, 2017.

[2] S. Cho, J. Wang, and S. Lee. Handling outliers in non-blind image deconvolution. In ICCV, 495–502, 2011.