Introduction to Computer Vision S21 Assignment #2

Due April 26/(Mon)

1. Weighted Giuded Filter

The original guided filter [1] uses a box window in the filter design. However, this causes a rotationally asymmetric response and does not reflect the spatial relationship. So, this can be remedied by adopting a Gaussian weighted window instead. In this case, the problem becomes to find a_k, and b_k that minimizes the following cost function in the window ω_k :

$$E(a_k, b_k) = \sum_{i \in \omega_k} \exp\left(-\|\mathbf{x}_i - \mathbf{x}_k\|^2 / \sigma_g^2\right)\left\{(a_k I_i + b_k - p_i)^2 + \varepsilon a_k^2\right\},$$

where p,q, and ε are the input image, output image, and regularization parameter, respectively, and **x** denotes a pixel position. The output can be obtained by

$$q_i = \bar{a}_i I_i + \bar{b}_i, \text{ where } \bar{a}_i = \frac{1}{|\omega|}\sum_{k \in \omega_k} a_k \text{ and } \bar{b}_i = \frac{1}{|\omega|}\sum_{k \in \omega_k} b_k .$$

Following is the Algorithm for the Guided filter.

Algorithm Guided Filter.
Input: filtering input image *p*, guidance image *I*, radius r, regularization ε
Output: filtering output *q*.

1: mean_*I* = *f*_mean (*I*)
   mean_*p* = *f*_mean (*P*)
   corr_*I* = *f*_mean (*I*.* *I*)
   corr_*Ip* = *f*_mean (*I*.* *p*)
2: var_*I* = corr_*I* – mean_*I*.* mean_*I*
   cov_*Ip* = corr_*Ip* - mean_*I*.* mean_*p*
3: *a* = cov_*Ip*./(var_*I* + ε)
   b = mean_*p* – a.* mean_*I*
4: mean_*a* = *f*_mean (*a*)
   mean_*b* = *f*_mean (*b*)
5: *q* = mean_*a*.**I* + mean_*b*

For applying a Gaussian window, you need to use a 2D Gaussian filter for *f*_mean () in the algorithm.

$$G(\mathbf{x}) = \frac{1}{2\pi\sigma^2}\exp\left(-\|\mathbf{x}\|^2 / \sigma^2\right)$$

- Write your program for the Gaussian weighted guided filtering algorithm.
- Given test noisy color images (afghan_noise#.png), convert them into grey images using $I = (R + G + B)/3$.
- Clean the grey images by your algorithm by setting $I = p$. How about setting I as one of R, G and B?
- Modify your algorithm for multichannel input and test it on the color images.
- Show the noisy images and filtered images with different parameter settings.
- Provide the best result, its PSNR value and the running time.

  $PSNR = 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right)$, where MAX_I is the maximum possible pixel value of the image,

  and $MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left[q(i,j) - \bar{q}(i,j)\right]^2$, $\bar{q}$:original clean image, q: denoised image
- [Extra Credit] Do you have any idea to improve the performance or enhance the running time?

2. Weighted Median Filter

The original median filter is to replace a pixel value by the median value of its neighboring pixels. The median can be computed from a histogram

$$h(\mathbf{x}, i) = \sum_{\mathbf{x}' \in N(\mathbf{x})} \delta\left(p(\mathbf{x}') - i\right),$$

where $N(\mathbf{x})$ is a local window near $\mathbf{x}$, p is the pixel value, i is the discrete bin index, and $\delta(\cdot)$ is the Kronecker delta function.

The weighted median filter is a modification of it in which the pixels are weighted in the local histograms such that

$$h(\mathbf{x}, i) = \sum_{\mathbf{x}' \in N(\mathbf{x})} W(\mathbf{x}, \mathbf{x}')\delta\left(p(\mathbf{x}') - i\right).$$

The weigh function $W(\mathbf{x}, \mathbf{x}')$ can be the box filter, bilateral filter, or guided filter.

- Now, implement the weighted median filters with different weight functions of
  i) Box filter
  ii) Gaussian filter
  iii) Bilateral filter [2]

- Test your algorithm on the noisy test (grey) images (monkey_noise#.png) with varying sizes of filters ($(r+1) \times (r+1), r = 2,4,8$), with appropriate parameters for each weight filter.
- Show and compare your results qualitatively and quantitatively.
- Report your best results with the PSNR, and the running time.
- [Extra Credit] Do you have any idea to improve the performance or enhance the running time?

Implementation & Submission instructions:

- Implementation instruction: You can use MATLAB, C++, or Python for your implementation.
- Submission instructions:
- Upload the electronic file that includes the report, source code, and data in a single zip format with the name "**ICV_assignment#2_yourname.zip**" on the ETL class homepage.
- The report should include a brief description of the problems, code, results, and discussions

- Reference

[1] Guided Image Filtering, by Kaiming He, Jian Sun, and Xiaoou Tang, in TPAMI 2013
[2] Bilateral Filtering for Gray and Color Images, by C. Tomasi and R. Manduchi in ICCV1998

Note: All works should be individual-based. NO copy is allowed.