digital image processing hw2

problem 1: guided filter

Guided filter [1] is an explicit image filter which has the edge-preserving smoothing property like the bilateral filter, but also handles gradient reversal artifacts within O(N) time complexity where N is the number of pixels.

In joint bilateral filter, input image can be smoothed with the help of a guidance image, resulting more reliable smoothing regarding the precise edge components. However, bilateral filter suffers from the gradient reversal artifacts because the pixels around the edge has high intensity variance which makes the averaged output unreliable. Furthermore, the naive implementation of the bilateral filter elapses O(Nr^2) time, which is impractical when kernel radius r is large.

In guided filter, filter output is assumed to be a linear transformation of guidance I with linear coefficients a_k, b_k in a window ω_k of center pixel k. Coefficients are optimized to minimize the difference between filter output q and input p:

$$E(a_k, b_k) = \sum_{i \in \omega_k} ((a_k I_i + b_k - p_i)^2 + \varepsilon a_k^2)$$

which can be solved by linear regression to have each coefficients as:

$$a_k = \frac{1/|\omega| \sum_{i \in \omega_k} I_i p_i - \mu_k \bar{p}_k}{\sigma_k^2 + \varepsilon}$$

$$b_k = \bar{p}_k - a_k \mu_k$$

where μ_k, σ_k^2 are the mean, variance of I in ω_k, $\bar{p}_k = 1/|\omega| \sum_{i \in \omega_k} p_i$. εa_k^2 is a regularization term to prevent explosion of a_k. Moreover, low-pass filter can be approximated when $\sigma_k^2 << \varepsilon$ and thus $a_k \sim 0$, $b_k \sim \bar{p}_k$.

When applying the window in the image, several windows ω_k can affect a same pixel i. Thus output q_i can be averaged from all the possible windows ω_k in which the pixel i is contained:

$$q_i = 1/|\omega| \sum_{k: i \in \omega_k} (a_k I_i + b_k)$$

From the equation above, it can be found that box filters ($\sum_{i \in \omega_k} f_i$) are the only filters applied in calculating the output image. Thus by using the integral image technique, the time complexity of guided filter becomes O(N) where N is the number of pixels.

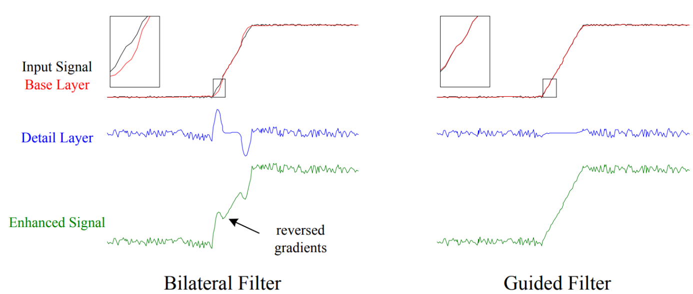

Figure 1: gradient reversal effect from bilateral filter and the corresponding guided filter result

Another good property of the above equation is that $\bigtriangledown q$ can be approximated as a scaling of $\bigtriangledown I$ only when big intensity change causes small value for $\bigtriangledown \bar{a}_i$, $\bigtriangledown \bar{b}_i$ where $\bar{a}_i = 1/|\omega| \sum_{k \in \omega_i} a_k$ and $\bar{b}_i = 1/|\omega| \sum_{k \in \omega_i} b_k$. By assuring scaling relation only on edge, unlike bilateral filter, which gets confused when trying to smooth edge component, gets stable on the edge and prevents gradient reversal problem. Visual representation of gradient preserving property in the guided filter can be found in Figure1. Actual result, Figure2, also shows that gradient reversal effect is diminished by applying guided filter rather than bilateral filter.

From the equation above, the output can be also defined as a linear transformation of the input.

$$q_i = \sum_j W_{ij}(I) p_j$$

Figure 2: detail-enhanced result of bilateral filter and guided filter. $\sigma_s = 16$, $\sigma_r = 0.1$ is used for bilateral filter, whereas $r = 16$ and $\varepsilon = 0.1^2$ is used for guided filter

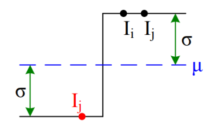

Figure 3: relation between W_{ij} and the directions of I_i, I_j

$$W_{ij}(I) = 1/|\omega|^2 \sum_{k:(i,j) \in \omega_k} \left(1 + \frac{(I_i - \mu_k)(I_j - \mu_k)}{\sigma_k^2 + \varepsilon}\right)$$

This form gives a good intuition about edge-preserving property of the guided filter; As in Figure3, when I_i and I_j are on the other side of an edge, then $(I_i - \mu_k)(I_j - \mu_k)$ becomes negative, thus $(1 + \frac{(I_i - \mu_k)(I_j - \mu_k)}{\sigma_k^2 + \varepsilon})$ gets close to zero, resulting small W_{ij} value. Semantically, small W_{ij} means that pixels positioned on p_i and p_j are not smoothed on a scale. Contrastly, when I_i and I_j are on the same side of an edge, $(I_i - \mu_k)(I_j - \mu_k)$ becomes positive, making W_{ij} reasonably big value to smooth the pixels positioned on p_i and p_j.

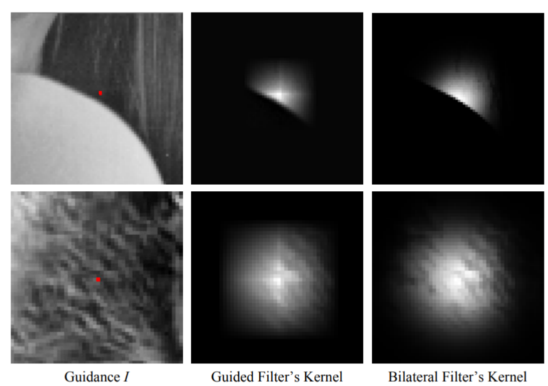

Figure 4: kernel values of guided filter/bilateral filter in various guide image

Figure4 contains kernel values of both guided filter and bilateral filter. On top guide image, it can be also seen that on both bilateral filter and guided filter, kernel is constructed as to preserve dominant edge (dividing bottom and top diagonally). Guidance image having less dominant edge incurs both bilateral filter and guided filter to have omnidirectional smoothing effect.

As discussed above, guided filter assumes linear transformation between guidance image and input image, then aggregates the guide-transformed output within a window. This can prevent the occurrence of gradient reversal problem with time complexity of O(N). Guided filter can be applied in various fields of image processing, including detail enhancement, denoising, matting, haze removal and so forth.

problem 1: rolling guidance filter

Rolling guidance filter [2] is a scale-aware local operation filter that controls smoothing details with a scale measure, by which small-scale structures can be removed while preserving others. Despite of its simple implementation, it can be run with various types of guidance filters, and converges within several steps of iteration.

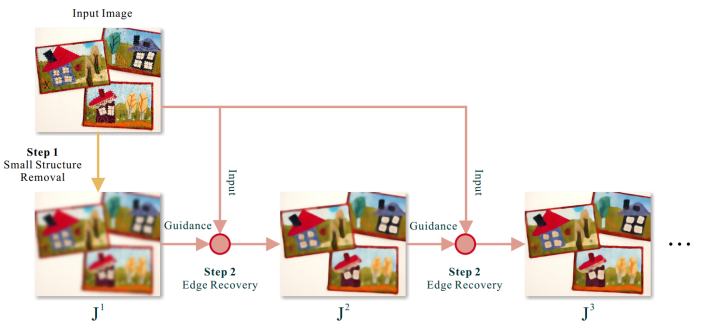

Figure 5: flow diagram of rolling guidance filter

Rolling guidance filter consists of two steps; small structure removal and edge recovery (Figure5). At beginning, gaussian filter is used to remove small structures, which also involves blurring at large-scale intensity variations.

$$G(p) = 1/K_p \sum_{q \in N(p)} exp(-\frac{||p-q||^2}{2\sigma_s^2})I(q)$$

$$K_p = \sum_{q \in N(p)} exp(-\frac{||p-q||^2}{2\sigma_s^2})$$

where K_p is the normalization term and $N(p)$ stands for the set of neighboring pixels of p. The next edge recovery step iteratively recovers edge with joint bilateral filter, with the guidance image as its previous step output. J^t stands for the recovery result from the t-th iteration. It can also be possible to use other types of edge-recovery filter such as guided filter.

$$J^{t+1}(p) = JointBilateral(p, I, J^t, \sigma_s, \sigma_r)$$

$$= 1/K_p \sum_{q \in N(p)} exp(-\frac{||p-q||^2}{2\sigma_s^2} - \frac{||J^t(p) - J^t(q)||^2}{2\sigma_r^2})I(q)$$

$$K_p = \sum_{q \in N(p)} exp(-\frac{||p-q||^2}{2\sigma_s^2} - \frac{||J^t(p) - J^t(q)||^2}{2\sigma_r^2})$$

When using joint bilateral filter as its guidance, the small structure removal step and the edge recovery step can be merged if J^0 is a constant-value image. Assign J^t as a constant C, then the resulting J^t has the form exactly same as gaussian filter. The combined iteration is described in Algorithm 1.

$$J^t(p) = C \forall p$$

$$J^{t+1}(p) = 1/K_p \sum_{q \in N(p)} exp(-\frac{||p-q||^2}{2\sigma_s^2} - \frac{||J^t(p) - J^t(q)||^2}{2\sigma_r^2})I(q)$$

$$= 1/K_p \sum_{q \in N(p)} exp(-\frac{||p-q||^2}{2\sigma_s^2})I(q) = G(p)$$

Algorithm 1: rolling guidance filter

$J^0 \leftarrow$ constant image;
for $t = 1 : num\_iter$ **do**
  $J^t \leftarrow joint\_bilateral(I, J^{t-1}, \sigma_s, \sigma_r)$;
end
return J^{num_iter};

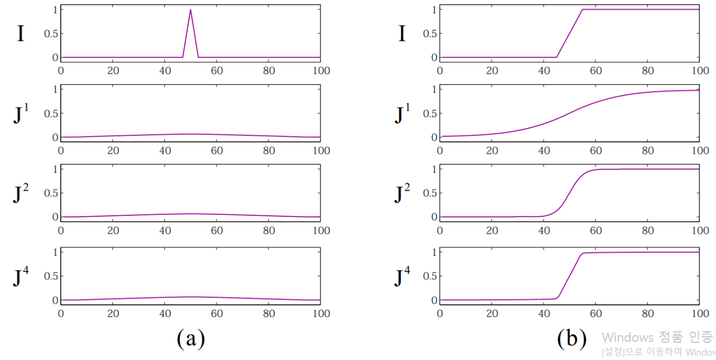

Figure 6: iteration result J^t in 1-D signal. (a) small structure (b) large structure, edge component

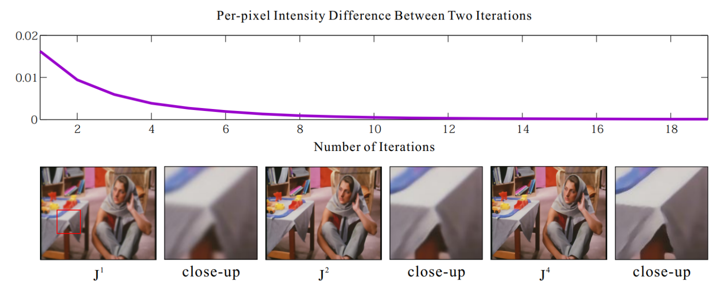

Figure 7: output image J_t on different timestep within an iteration

The detailed iteration process, its output in 1-D signal example and the real image can be found from Figure6, Figure7 respectively. As the iteration progresses, small structure smooths out (I to J_1 from Figure6 (a)) while large structure gets blurred at the first phase and then the edge is recovered. Intermediate results on iterative edge recovery phase (Figure7) also indicate the average number of iterations needed for the final output. It can be seen that in the specific image, within 10 iterations the edge is almost recovered.

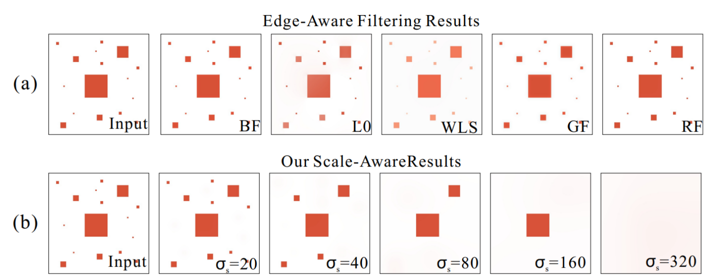

Figure 8: comparison on edge-aware filtering (bilateral, L^0, WLS, guided, recursive filters) and scale-aware filtering (rolling guidance filter)

Figure8 shows the scale-aware nature of the rolling guidance filter. It can be seen that the existing edge-aware filterings cannot successively remove the small scale elements while preserving the large scale element as it is, while in rolling guidance filter depending on σ_s the elements are masked out in the order of their scales.

Due to its simple implementation with fast and reasonably good result on scale-aware filtering, many applications can earn benefit from the rolling guidance filter; including detail enhancement, tone mapping, multi-scale structure decomposition, segmentation and so on. Especially, virtual contour restoration characteristic of the filter can boost performance on segmentation task by providing large-scale image information.

problem 2: cross image filtering

hw2_2.m reads input, guidance image and apply both joint bilateral filter and guided filter. input image, art_depth_low_res.png contains low resolution depth image, and guidance image, art_color_high_res.png contains high resolution color image. The goal of applying the filters onto the images is to increase edge details within the depth image by utilizing color image as its guidance.

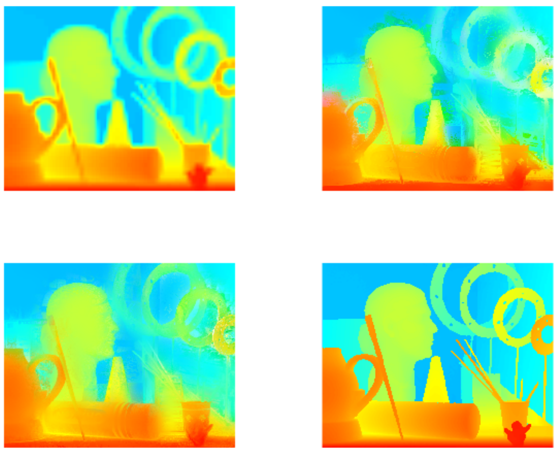

Figure 9: result of hw2_2.m. top left : original input image, top right : output of joint bilateral filter with kernel diameter = 65, σ_s, σ_r = 10, bottom left : output of guided filter with kernel diameter = 65, $\varepsilon = 4 * 10^{-1}$, bottom right : ground truth high resolution image

Figure9 shows the result of applying both filters given input and guidance image. It can be seen that in both joint bilateral and guidance filter, the details on edge components are boosted. However, no discrete segmentation could be achieved like in the groundtruth image from both filters. That would be partially due to the intensity variance of the guidance image. Flattening guidance image to have less textual or small scaled variation might help in the case.

Pseudocode for joint bilateral and guided filters are as follows. It can be seen that the complexity of joint bilateral filter and guided filter is O(Nr^2) and O(N) respectively where N is the size of input image and r is kernel radius (In the pseudocode of guided filter, extracting mean from each window is described; vanilla implementation will require O(r^2) computation, but this can be achieved in constant time by the integral technique):

Algorithm 2: joint_bilateral.m

$pad_i \leftarrow pad(input, radius)$;
$pad_g \leftarrow pad(guidance, radius)$;
$gauss\_range \leftarrow gauss(radius, \sigma_s)$;
for $i = 1 : M$ **do**
  for $j = 1 : N$ **do**
    $window_i \leftarrow pad\_input(i : i + 2 * radius, j : j + 2 * radius)$;
    $window_g \leftarrow pad\_guidance(i : i + 2 * radius, j : j + 2 * radius)$;
    $gauss\_spatial \leftarrow exp(-(window_i(i,j) - guidance(i,j))^2/(2 * \sigma_r^2))$;
    for $k = 1 : C$ **do**
      $denom \leftarrow sum(gauss\_range * gauss\_spatial)$;
      $output(i,j,k) \leftarrow sum(gauss\_range * gauss\_spatial * window_i)/denom$;
    end
  end
end

Algorithm 3: guided.m

pad $input$, $guidance$ into pad_i, pad_g;
calculate μ_g, σ_g;
$a, b \leftarrow zeros$;
for $i = 1 : M$ **do**
  for $j = 1 : N$ **do**
    for $k = 1 : C$ **do**
      extract $window_g$ and $window_i$;
      $a(i,j,k) = mean(window_g * window_i - \mu_g(i,j,k) * mean(window_i))$;
      $a(i,j,k) = a(i,j,k)/(\sigma_g(i,j,k) + \varepsilon)$;
      $b(i,j,k) = mean(window_i) - a(i,j,k) * \mu_g(i,j,k)$;
    end
  end
end
pad a, b into pad_a, pad_b;
for $i = 1 : M$ **do**
  for $j = 1 : N$ **do**
    for $k = 1 : C$ **do**
      extract $window_a$ and $window_b$;
      $output(i,j,k) = mean(window_a * guidance(i,j,k) + window_b)$;
    end
  end
end

problem 3: texture removal

hw2_3.m reads image.png, apply rolling guidance filter with both joint bilateral filter and guided filter. The result can be found in Figure10.

In Figure10, the leftmost images on both top and bottom are result of applying gaussian filter on the original image. It can be seen that both small structures like texture of table are smoothed out while large structures like table corner are blurred. As iteration proceed, large scale edge components are gradually restored.

problem 4: WLS filter

hw2_4.m reads noisy_image.png then apply bilateral, guided and WLS filter. Figure11 depicts the result of applying various filters onto the noisy input. It can be seen that all three filters can remove noise in some amount, but their edge preserving status are different. In bilateral filter output, edges are blurred in great amount. Though this edge smoothing effect is alleviated on guided filter, it also contains some blurring effect. On WLS result, while suppressing noisy component, most of edge components are successively preserved.

Pseudocode for WLS filter implementation is as below. Efficient calculation of Laplacian matrix could be achieved by the sparse matrix.

Algorithm 4: wls.m

$dl\_dx \leftarrow diff(log(input), 1, 2)$;
$a_x \leftarrow 1/(abs(dl\_dx)^\alpha + \varepsilon)$;
flatten a_x;
$a_x\_shift \leftarrow shift(a_x, M)$;
$dl\_dy \leftarrow diff(log(input), 1, 1)$;
$a_y \leftarrow 1/(abs(dl\_dy)^\alpha + \varepsilon)$;
flatten a_y;
$a_y\_shift \leftarrow shift(a_y, 1)$;
$L_g\_diag \leftarrow a_x + a_x\_shift + a_y + a_y\_shift$;
build sparse diagonal matrix L_g with diagonal element $-a_x$, $-a_y$ and L_g_diag;
$lhs \leftarrow I + \lambda * L_g$;
solve $lhs * u = input$ linear equation;

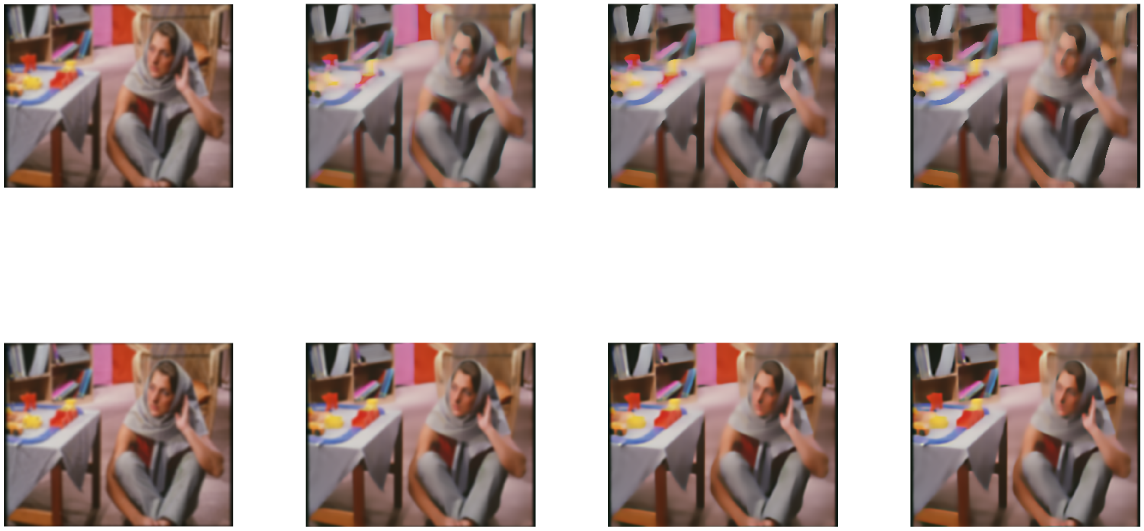

Figure 10: result of hw2_3.m. top : joint bilateral filter as guidance for rolling guidance, bottom : guided filter as guidance for rolling guidance. From left to right, each image stands for J^1, J^2, J^4 and J^6 intermediate output

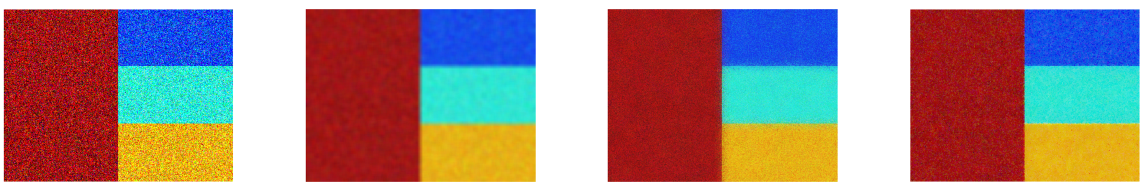

Figure 11: result of hw2_4.m. left : original image, second : output of bilateral filter with kernel size = 5, σ_s = 5, third : output of guided filter with guidance as itself, kernel diameter = 17 and $\varepsilon = 10^{-1}$, fourth : output of WLS filter with $\lambda = 2$, $\alpha = 1.2$ and $\varepsilon = 10^{-4}$

references

[1] He, Kaiming, Jian Sun, and Xiaoou Tang. "Guided image filtering." IEEE transactions on pattern analysis and machine intelligence 35.6 (2012): 1397-1409.

[2] Zhang, Qi, et al. "Rolling guidance filter." European conference on computer vision. Springer, Cham, 2014.