Propagated Filter

- Jen-Hao Rick Chang
- Yu-Chiang Frank Wang
- CVPR Paper 2015

Team - Rebooting
- Vidit Jain (201501021)
- Nakul Vaidya (201501108)
- Aayush Tiwari (201531031)

Guidance : Dr. Avinash Sharma
TA : Aaron Varghese

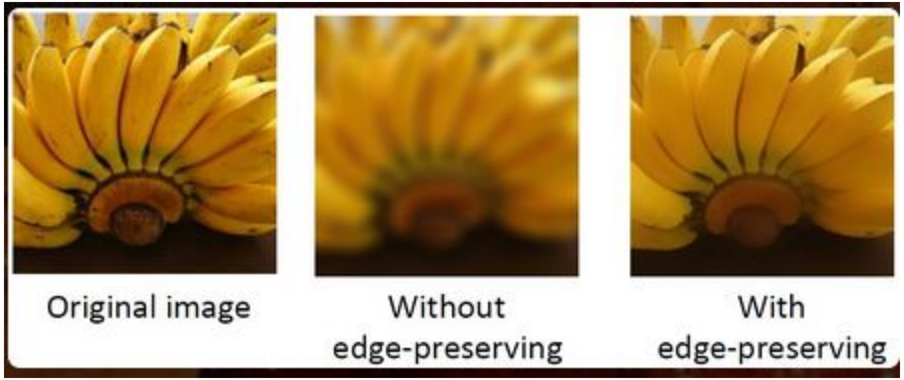
Introduction

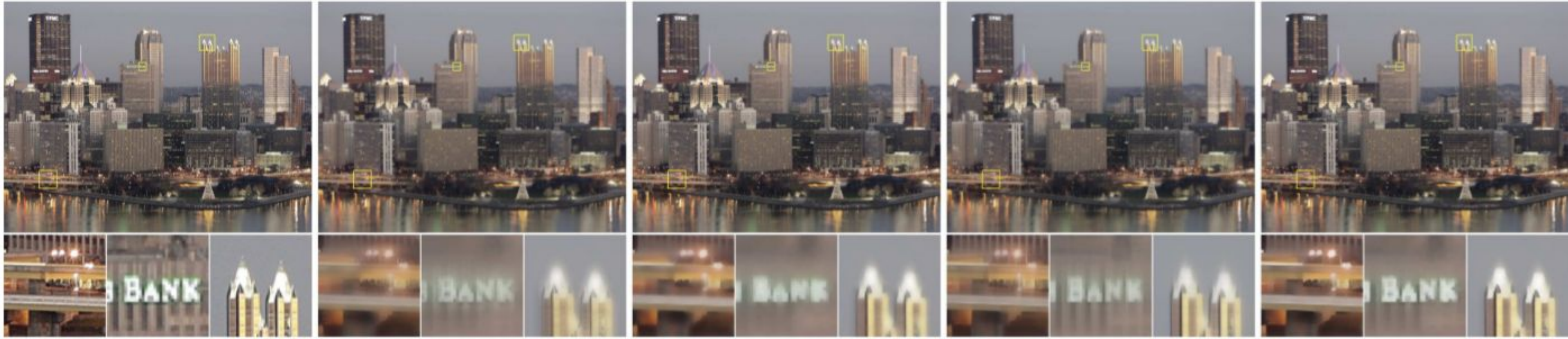
- Image filtering is a process of updating pixel values in an image to achieve particular goals like denoising, smoothing, enhancement, or matting.
- Bilateral and guided filters are popular edge-preserving image filters.
- However, these filters require predefined pixel neighborhood regions (via spatial functions or kernels), which are typically difficult to determine beforehand.
- Geodesic filters which dynamically determines their filtering kernels suffer from cross-region mixing in smoothing and other filtering tasks.
- Hence we need a filter for solving the above tasks.

What to design

- Come up with a filter that is able to observe and preserve image characteristics without the need to apply explicit spatial kernel functions.
- Cross-region mixing is a scenario where the characteristics of adjacent image regions are blended, the output image would contain blurry regions which result in degraded visual quality.
- So our filter also handles the problem of cross-region mixing.

What to achieve





Original Image Bilateral-Cut Filter Geodesic Filter Recursive Bilateral **Propagation Filter**

Other Filters

- → Bilateral Filtering
- → Guided Filtering
- → Geodesic Filtering

Bilateral Filtering

- A bilateral filter is a non-linear, edge-preserving, and noise-reducing smoothing filter for images.
- Bilateral filter replaces the intensity of each pixel with a weighted average of intensity values from nearby pixels.
- This weight can be based on a Gaussian distribution. Crucially, the weights depend not only on Euclidean distance of pixels, but also on the radiometric differences (e.g., range differences, such as color intensity, depth distance, etc.). This preserves sharp edges.

$$I'_s = \frac{1}{Z_s} \sum_{t \in \Omega} g(d_{\mathrm{BF}}(s,t); \sigma_s)\, g(d_{\mathrm{BF}}(I_s, I_t); \sigma_r)\, I_t,$$

Bilateral Filtering(continued)

Where I'_s is the filtered output at pixel s, Ω denotes the set of pixels t in the input I, and g(x; σ) is a Gaussian function with variance σ^2 . For bilateral filters, the spatial and photometric distances between pixels s and t are defined as:

$$d_{\mathrm{BF}}(s,t) = ||t - s||, \text{ and } d_{\mathrm{BF}}(I_s, I_t) = ||I_t - I_s||,$$

which calculate the Euclidean distance between their locations and that between their pixel values, respectively.

Guided Filtering

- The guided filter computes the filtering output by considering the content of a guidance image, which can be the input image itself or another different image denoted by I^g.
- It assumes that the filtered output at pixels is a linear transformation of the pixels(within a window W_k) of I^g.
- The guided filter can be used as an edge-preserving smoothing operator like the popular bilateral filter, but it has better behaviors near edges.
- The guided filter is controlled by two parameters the window radius, the extent of smoothing .

- The output image can be represented as $I'_s = \sum_{t \in N(s)} w^g_{s,t} I_t$

Where,

$$w^g_{s,t} = \frac{1}{|W|^2} \sum_{k \in \{k|s,t \in W_k\}} \left(1 + \frac{\left(I^g_s - \mu_k\right)\left(I^g_t - \mu_k\right)}{\sigma^2_k + \epsilon}\right).$$

|W| : window size

“μ_k” and”σ_k^2” are the mean and variance of pixel values in window W_k of I^g, respectively.

Geodesic Filtering

- Geodesic filtering utilizes only photometric distances during the filtering process.
- It calculates the output at pixels by $I'_s = \frac{1}{Z_s} \sum_{t \in \mathcal{N}(s)} g(d_{\mathrm{GF}}(I_s, I_t); \sigma_r)\, I_t$

  Where Z_s is the normalization factor, and $N(s)$ contains neighboring pixels of s.

- The photometric distance between pixels s and t is defined as

$$d_{\mathrm{GF}}(I_s, I_t) = \min_{\phi} \sum_{x, x+1 \in \phi} \|I_{x+1} - I_x\|$$

  Where $\varnothing$ is the path connecting pixels s and t.

Propagated Filter

- Propagated filter suppresses undesirable information from adjacent or neighboring pixels during filtering by aiming at taking the context information between image pixels into consideration.
- Without using any explicit spatial functions, propagated filter is formed from a probabilistic point of view.
- Propagation filter essentially cooperates the merits of bilateral and geodesic filtering, while comparable computation costs can be obtained.

One-Dimensional Case

- Given an input image I , the filtered output I s ′ at pixel s produced by our propagation filter is calculated as shown in the formula.

The output pixels s is given as

$$I'_s = \frac{1}{Z_s} \sum_{t \in \mathcal{N}(s)} w_{s,t} \, I_t,$$

where $Z_s = \sum_{t \in \mathcal{N}(s)} w_{s,t}$ as a normalized factor

Finding $w_{s,t}$

- Definition 1
  - Suppose there are n singly connected pixels 1, . . . , n , in which s and t are two pixels satisfying $1 \leq s < t \leq n$ without the loss of generality. Pixel t is related to pixel s , or $s \rightarrow t$, if and only if $s \rightarrow t-1$, $t-1 \xrightarrow{a} t$, and $s \xrightarrow{r} t$. In addition, each pixel is always self-related, i.e., $s \rightarrow s$.

Finding $w_{s,t}$

- Definition 2
  - Suppose there are n singly connected pixels, 1, . . . , n , and pixels s and t satisfying 1 ≤ s ≤ t ≤ n . The weight ws,t for filtering pixel s with pixel t is the probability value of t being related to s , i.e., P (s → t) .

If $t = s$, we have $w_{s,s} = P(s \to s) = 1$. As for $t \neq s$, based on Definitions 1 and 2, we calculate the weight $w_{s,t}$ by the Bayes' rule:

$$\begin{aligned} w_{s,t} &\equiv P(s \to t) = P\Big(s \to t-1 \ \wedge\ t-1 \xrightarrow{a} t \ \wedge\ s \xrightarrow{r} t\Big) \\ &= P(s \to t-1)\ P\Big(t-1 \xrightarrow{a} t \ \wedge\ s \xrightarrow{r} t \,|\, s \to t-1\Big) \\ &= w_{s,t-1}\ P\Big(t-1 \xrightarrow{a} t \,|\, s \to t-1\Big)\ P\Big(s \xrightarrow{r} t \,|\, s \to t-1 \wedge t-1 \xrightarrow{a} t\Big) \\ &\equiv w_{s,t-1}\ D(t-1, t)\ R(s, t)\,. \end{aligned} \tag{7}$$

Finding $w_{s,t}$

R(s, t) is measured as the adjacent-photometric relationship between s and t . As a result, we define

$$R(x,y) = g(\|Ix - Iy\|; \sigma r) = \exp\left(\frac{-\|Ix - Iy\|^2}{2\sigma_r^2}\right)$$

For simplicity, we choose $\sigma_a = \sigma_r$, and thus D(·) = R(·) .

Finding $w_{s,t}$

The probability value of two adjacent pixels being photometric related is proportional to the value of a Gaussian function of their pixel value difference, we define

$$D(x,y) = g(\|I_x - I_y\|;\sigma_a) = \exp\left(\frac{-\|I_x - I_y\|^2}{2\sigma_a^2}\right)$$

where I_x is the value of pixel x , and $\| I_x - I_y \|$ measures the Euclidean distance between the corresponding pixel value difference.

Illustration for finding $w_{s,t}$

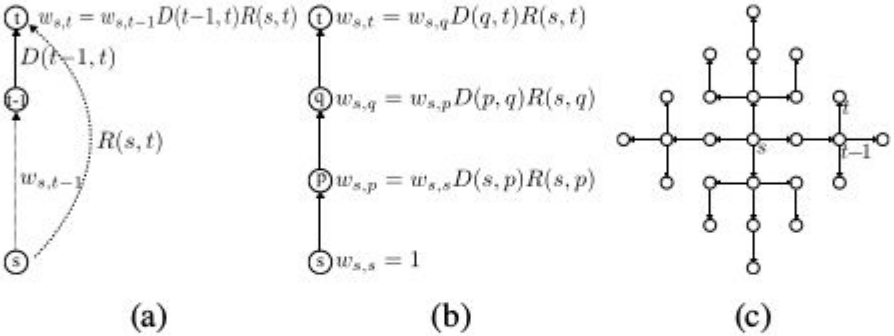
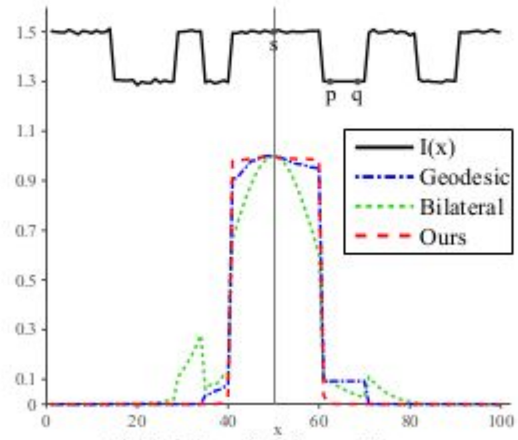


Figure 2: Illustration of propagation filtering. (a) the definition of filtering weight $w_{s,t}$, (b) the calculation of $w_{s,t}$, and (c) the pattern for performing 2D filtering with $d = 3$ pixels.

Observations

- Based on the above definitions, we see that a large weight $w_{s,t}$ not only needs pixels s and t to have a strong photometric relationship (i.e., similar pixel values), it also needs large $w_{s,t-1}$ and $D(t-1, t)$ values for the intermediate pixels between s and t .
- That means, if any pixel along the path connecting pixels s and t is unrelated to either of them, t will be viewed as unrelated to s . In other words, a small $w_{s,t}$ will be resulted.
- This is the reason why this propagation filter is able to reflect image context information when performing filtering.

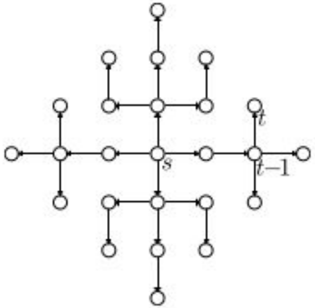
Comparison



(a) Weights of different filters

Two Dimensional Case

- We consider a particular 2D pattern for filtering all pixels in an image shown below
- The path would be a straight line connecting pixels s and t, if these two pixels are horizontally or vertically aligned.
- If pixels s and t are not simply horizontally or vertically connected, we determine the path based on their Manhattan distance.
- If the Manhattan distance between s and t is an odd number, we choose the path for traversing from that pixel to its predecessor in the vertical direction; otherwise, the horizontal path will be chosen.
- By using this 2D pattern for filtering, the computation complexity of our propagation filter will be reduced to O(NW).



Comparison to Bilateral and Geodesic Filtering

In propagation filters the filter weights are derived from a probability point of view. the filtering weights $w_{s,t}$ regarding the pixel distances as:

$$w_{s,t} = g\big(d^a_{\mathrm{PF}}(I_s, I_t);\ \sigma_a\big)\ g\big(d^r_{\mathrm{PF}}(I_s, I_t);\ \sigma_r\big)$$

$$d^a_{\mathrm{PF}}(I_s, I_t) = \sqrt{\sum_{x, x+1 \in \phi} \|I_{x+1} - I_x\|^2}$$

$$d^r_{\mathrm{PF}}(I_s, I_t) = \sqrt{\sum_{x \in \phi} \|I_x - I_s\|^2},$$

Note that φ is the path connecting pixels s and t .

Comparison to Bilateral and Geodesic Filtering

- Instead of using explicit spatial kernels, we have $d^a_{PF}(I_s, I_t)$ utilize intermediate adjacent pixel information between s and t , which is similar to the photometric distances applied in geodesic filtering and recursive bilateral filtering.
- The deficiency of geodesic filters is complemented by $d^r_{PF}(I_s, I_t)$, which measures the photometric distance between pixels along the path of interest.
- Geodesic filters cannot suppress effects from nearby image regions well, especially when only negligible noise is observed in those regions, Instead, our propagation filter is able to observe large $d^a_{PF}(I_s, I_t)$ and $d^r_{PF}(I_s, I_t)$ and have $w_{s,q} < w_{s,p}$.
- This would alleviate cross-region mixing problems, and this is the reason why propagation filters better discriminate between image regions with different context information.

Propagation filtering as Robust Estimator

- Propagation filter can be viewed as a one-step estimator, which aims at minimizing the error between the filtered and desirable outputs.
- We first transform our filtering algorithm of One-Dimensional into the following formulation:

$$I'_s = \frac{1}{Z_s} \sum_{t \in \mathcal{N}(s)} w_{s,t} I_t = I_s - \frac{1}{Z_s} \sum_{t \in \mathcal{N}(s)} w_{s,t} (I_s - I_t).$$

- The above equation is effectively a gradient descent solver optimizing the objective function f, whose gradient at pixel s is derived as: $$\nabla f = \sum_{t \in \mathcal{N}(s)} w_{s,t} \, (I_s - I_t) \, .$$
- With this gradient solver, the optimization problem can be recovered as:

$$\min_I \sum_{s \in \Omega} \sum_{t \in \mathcal{N}(s)} \int w_{s,t} \, (I_s - I_t) \, \mathrm{d}(I_s - I_t),$$

Where Ω denotes the pixel set of the input image.

Application and Results

Denoising

Original

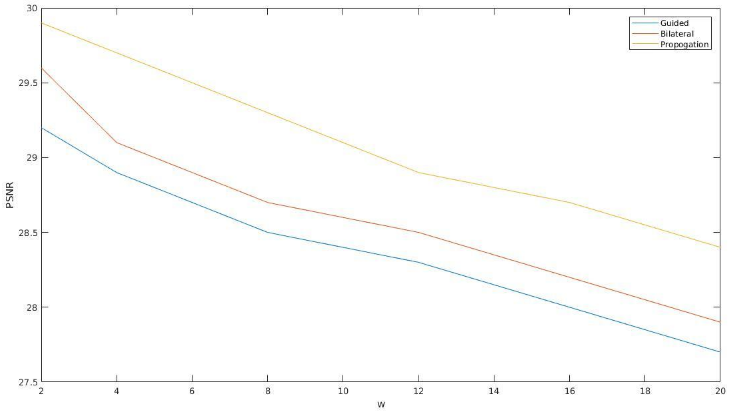
White Noise

Guided Filter

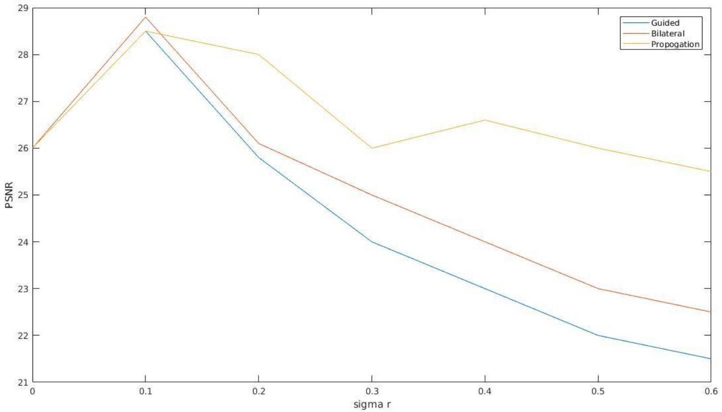
Bilateral Filter

Propagation Filter

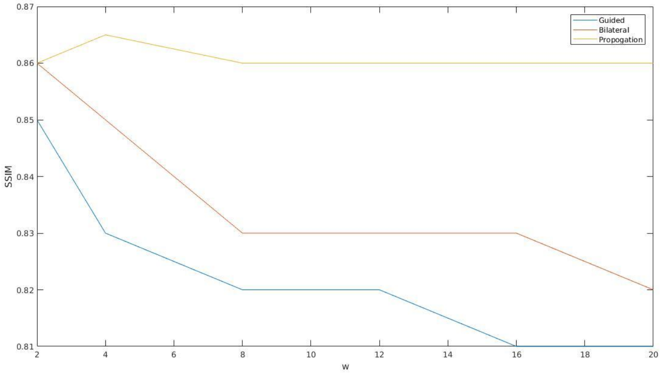
($\sigma_s = 5, \sigma_r = 40$)



PSNR vs w



PSNR vs sigma r



SSIM vs w

Original

White Noise

Bilateral Filter

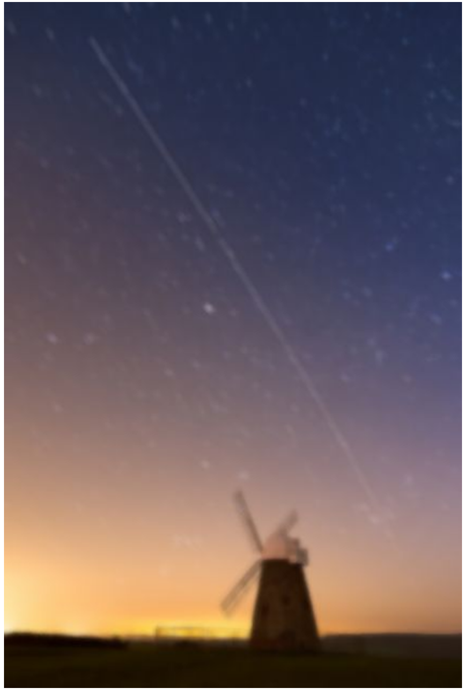
Guided Filter

Propagation Filter

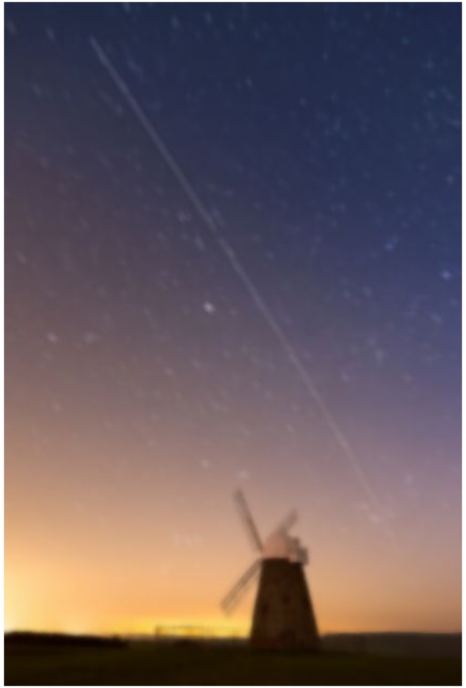
($\sigma_s = 5, \sigma_r = 40$)

Image Smoothing

Original Image



bilateral filter
sigma(s) = 5, sigma(r) = 0.4



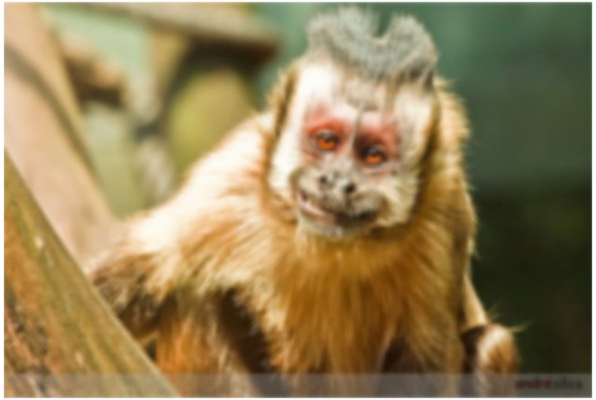
Guided filter
w = 5, sigma(r) = 0.4

Propagation filter
sigma(s) = 5, sigma(r) = 0.4

Image Smoothing

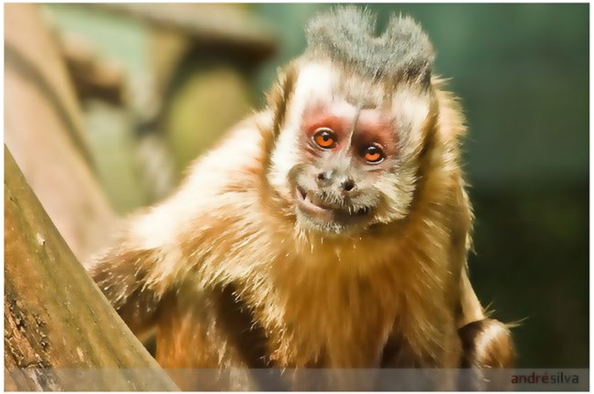
Original Image

bilateral filter, sigma(s) = 5, sigma(r) = 0.4



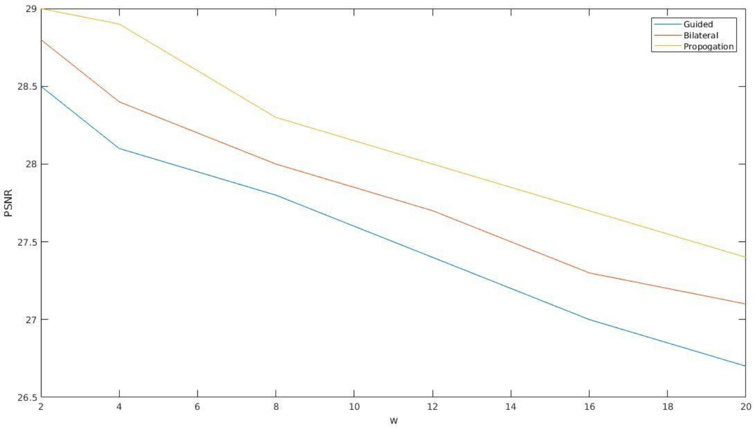
Guided filter

w = 5, sigma(r) = 0.4

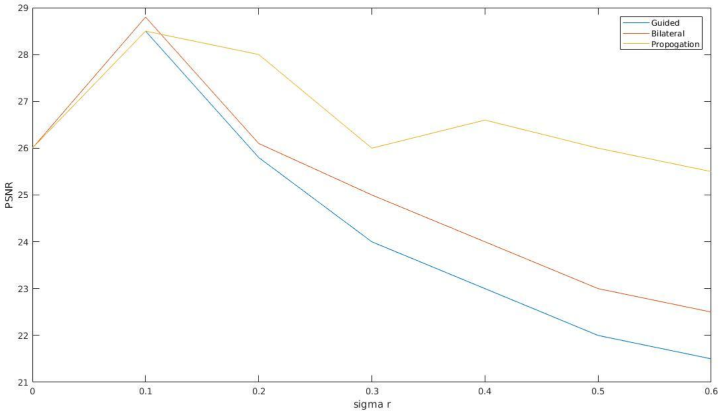


Propagation filter

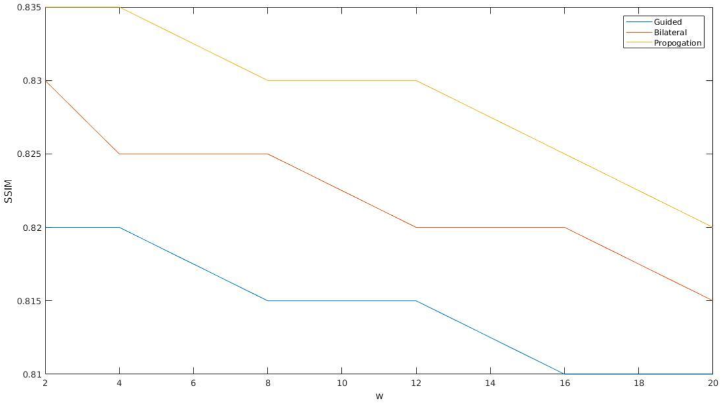
sigma(s) = 5, sigma(r) = 0.4



PSNR vs w



PSNR vs sigma r



SSIM vs w

Flash/No-flash Denoising

Flash

No Flash

Flash/No-flash Denoising

Guided Filter
w = 5, sigma(r) = 0.1

Propagation Filter
w = 5, Sigma(r) = 0.1

Flash/No-flash Denoising

Flash

No Flash

Flash/No-flash Denoising

Guided Filter
w = 5, sigma(r) = 0.1

Propagation Filter
w = 5, Sigma(r) = 0.1

Thank You!