Denoise:

Non-linear filtering

Dr. Tushar Sandhan

Introduction

- Image noise
  - a random variation of pixel values (RGB)
  - causes: sensor (quality, heat), camera parameters, environment, read noise

Introduction

- Image noise
  - a random variation of pixel values (RGB)
  - causes: sensor (quality, heat), camera parameters, environment, read noise

impulse

Introduction

- Image noise
  - a random variation of pixel values (RGB)
  - causes: sensor (quality, heat), camera parameters, environment, read noise

impulse

diffused impulses

Introduction

- Image noise
  - a random variation of pixel values (RGB)
  - causes: sensor (quality, heat), camera parameters, environment, read noise

impulse

diffused impulses

imperceptible

Introduction

- Image noise
  - a random variation of pixel values (RGB)
  - causes: sensor (quality, heat), camera parameters, environment, read noise

impulse

diffused impulses

imperceptible

salt & pepper

Gaussian filter

- Spatial averaging (linear) filter
  - noise looks high freq, so LPF
  - local neighbourhood of pixels has roughly similar color
  - while averaging, central pixels are weighted higher than far pixels
  - weights are fixed & don’t depend on image content

Gaussian filter

- Spatial averaging (linear) filter
  - noise looks high freq, so LPF
  - local neighbourhood of pixels has roughly similar color
  - while averaging, central pixels are weighted higher than far pixels
  - weights are fixed & don't depend on image content

$$G[I]_{\mathbf{p}} = \sum_{\mathbf{q} \in \mathcal{S}} G_{\sigma}(\|\mathbf{p} - \mathbf{q}\|)\, I_{\mathbf{q}},$$

Gaussian filter

- Spatial averaging (linear) filter
  - noise looks high freq, so LPF
  - local neighbourhood of pixels has roughly similar color
  - while averaging, central pixels are weighted higher than far pixels
  - weights are fixed & don't depend on image content

$$G[I]_{\mathbf{p}} = \sum_{\mathbf{q} \in \mathcal{S}} G_\sigma(\|\mathbf{p} - \mathbf{q}\|)\, I_{\mathbf{q}},$$

$$G_\sigma(x) = \frac{1}{2\pi\,\sigma^2}\, \exp\left(-\frac{x^2}{2\,\sigma^2}\right)$$

Gaussian filter

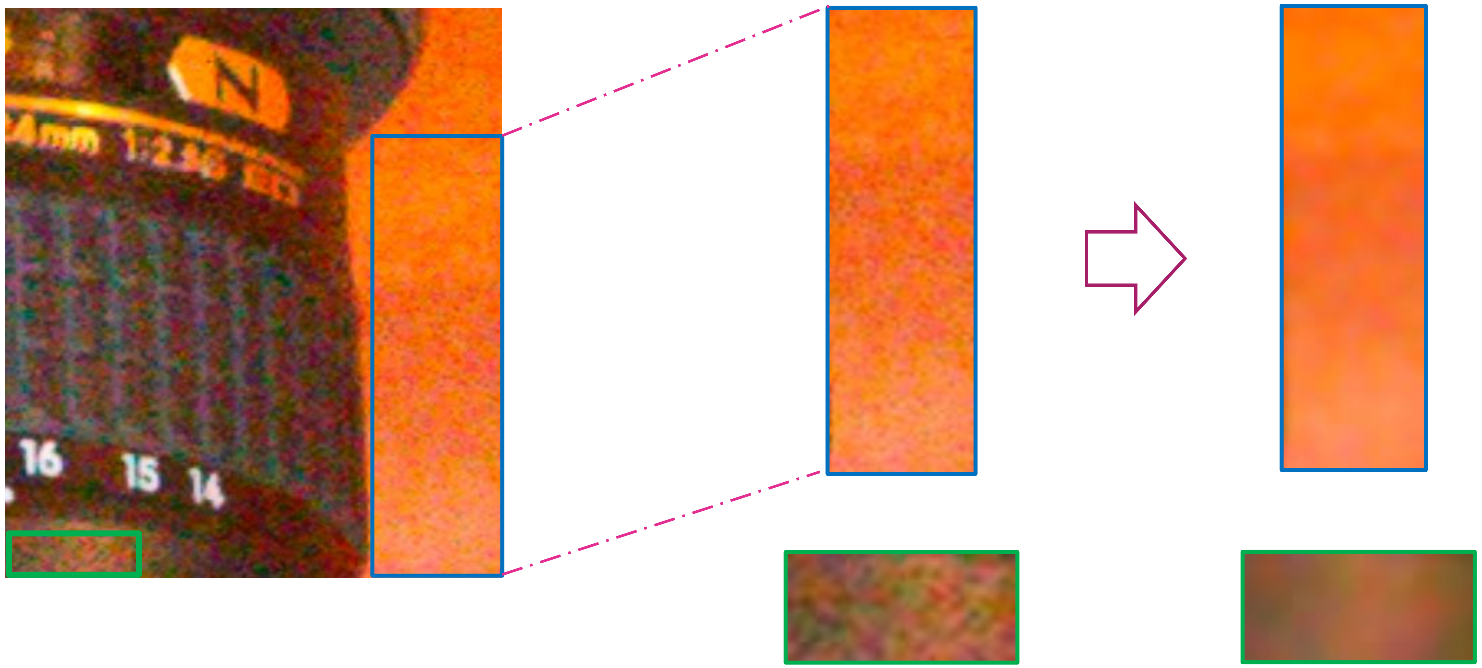
- Spatial averaging (linear) filter
  - noise looks high freq, so LPF
  - local neighbourhood of pixels has roughly similar color
  - while averaging, central pixels are weighted higher than far pixels
  - weights are fixed & don't depend on image content

$$G[I]_{\mathbf{p}} = \sum_{\mathbf{q}\in\mathcal{S}} G_\sigma(\|\mathbf{p}-\mathbf{q}\|)\, I_{\mathbf{q}},$$

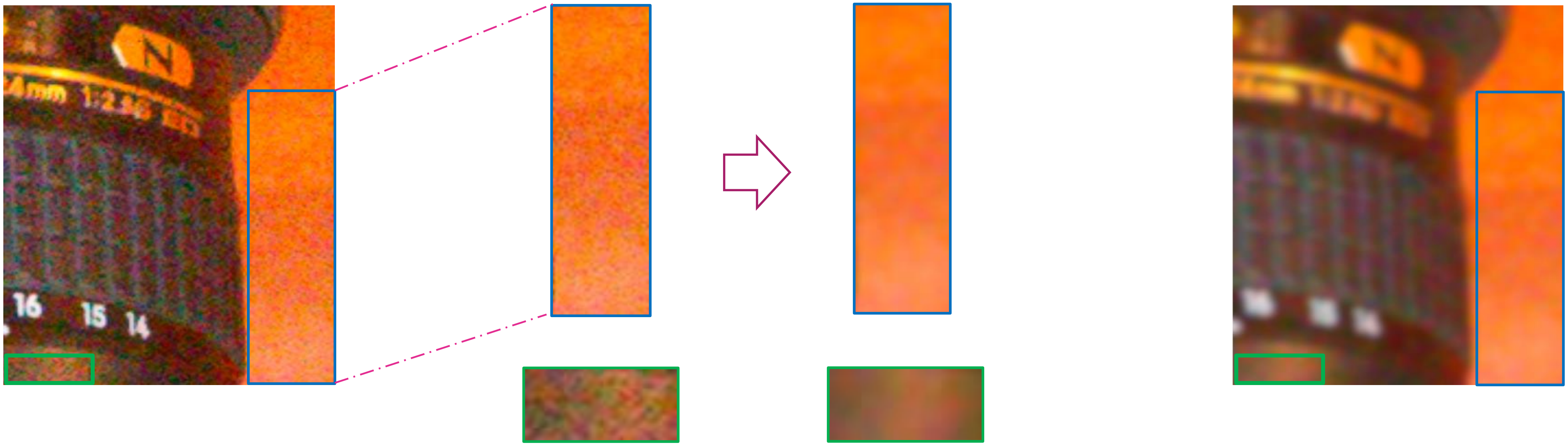
$$\frac{1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

$$G_\sigma(x) = \frac{1}{2\pi\,\sigma^2}\,\exp\left(-\frac{x^2}{2\,\sigma^2}\right)$$

Gaussian filter



Gaussian filter



Unsharp masking

- Fill back the lost edges
  - an image sharpening method
  - amplifies high freq.

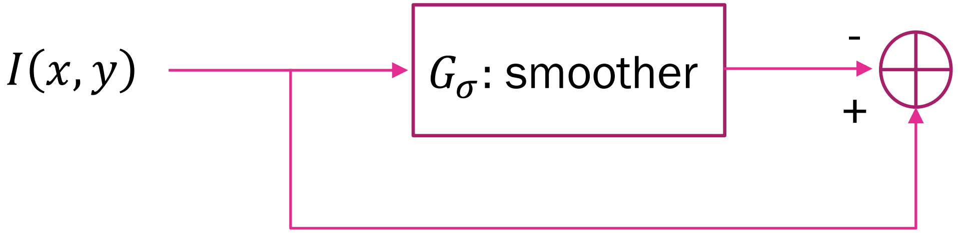
$$f_{sharp}(x,y) = I(x,y) + \alpha * \Big(I(x,y) - G_\sigma\big(I(x,y)\big)\Big)$$

$$I(x,y) \longrightarrow \boxed{G_\sigma\text{: smoother}} \longrightarrow$$

Unsharp masking

- Fill back the lost edges
  - an image sharpening method
  - amplifies high freq.

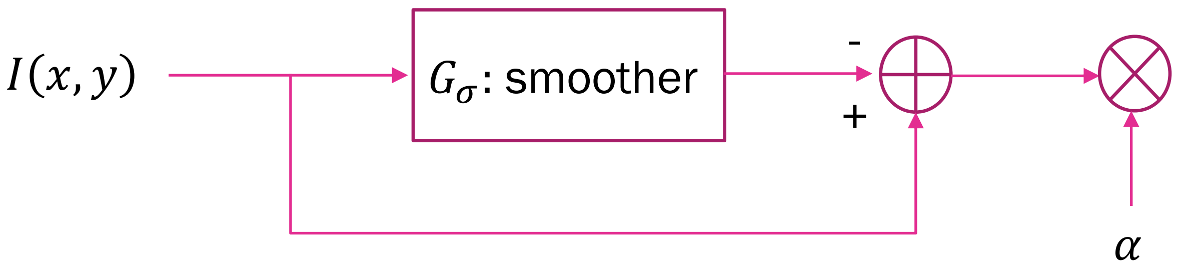
$$f_{sharp}(x,y) = I(x,y) + \alpha * \left(I(x,y) - G_\sigma\big(I(x,y)\big)\right)$$



Unsharp masking

- Fill back the lost edges
  - an image sharpening method
  - amplifies high freq.

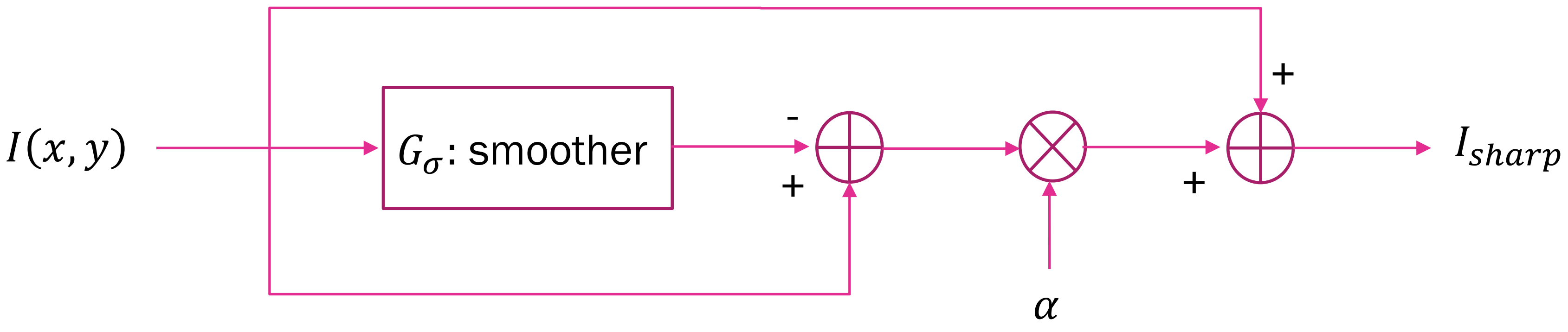
$$f_{sharp}(x,y) = I(x,y) + \alpha * \left(I(x,y) - G_\sigma\big(I(x,y)\big)\right)$$



Unsharp masking

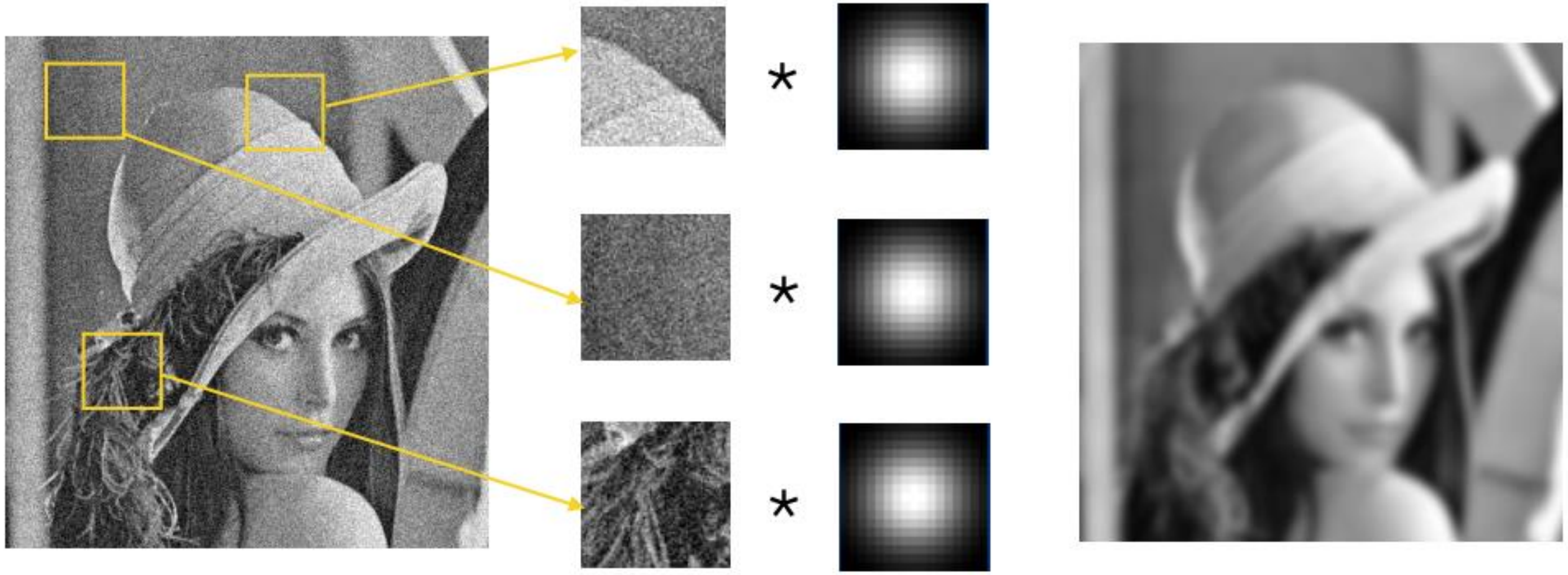
- Fill back the lost edges
  - an image sharpening method
  - amplifies high freq.

$$f_{sharp}(x,y) = I(x,y) + \alpha * \Big(I(x,y) - G_\sigma\big(I(x,y)\big)\Big)$$



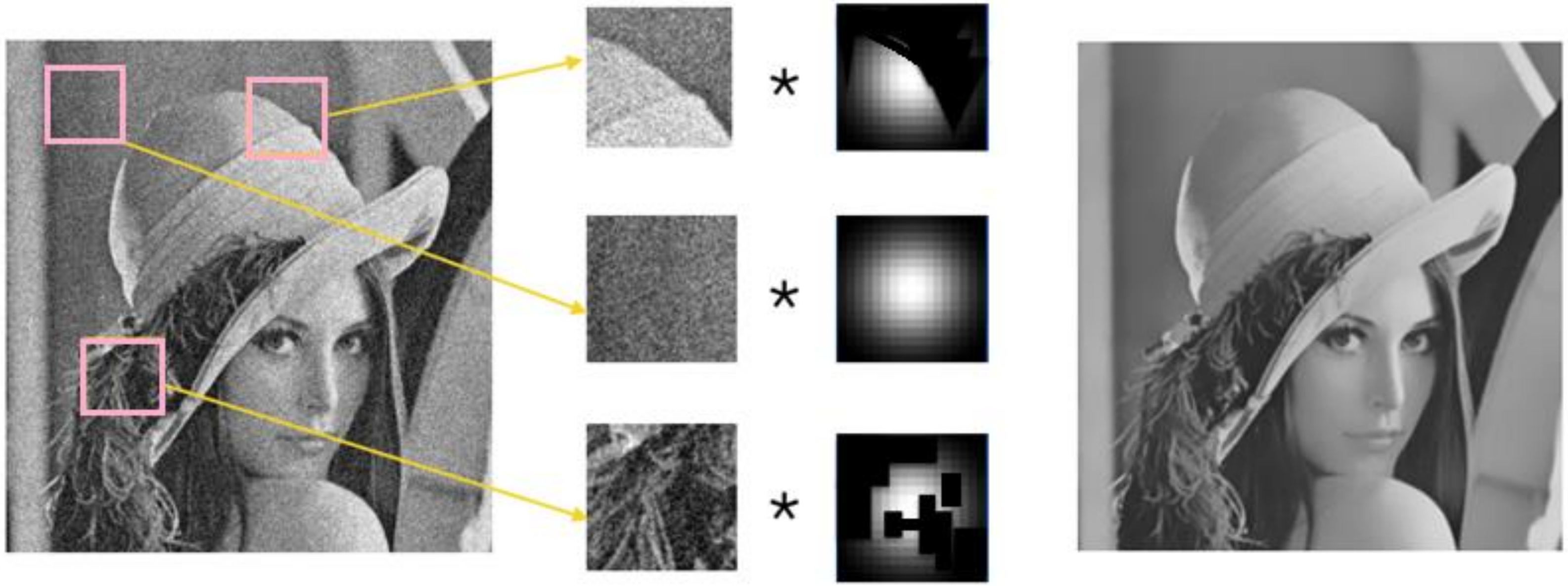
Gaussian filter

- Fixed kernel everywhere
  - edges are lost
  - averaging across edges



Dynamic filter

- Non-fixed kernel everywhere
  - edges are not lost
  - no averaging across edges
  - non-linear

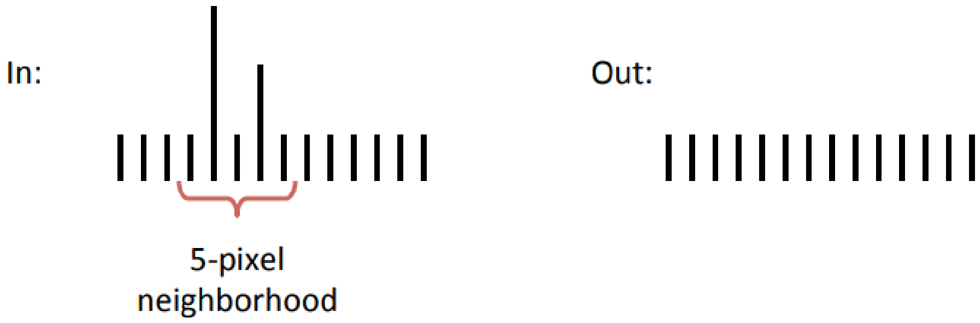


Median filter

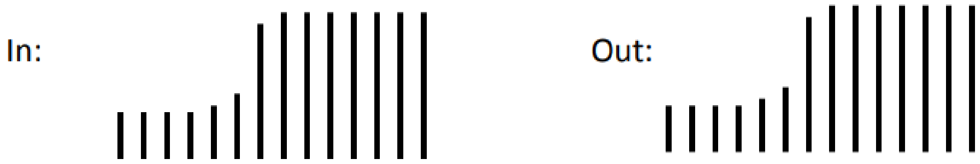
- Non-linear filter
- Replace each pixel by the median over a range N (e.g. $N = 5$)

Median([1 7 1 5 1]) = 1

- Spike noise



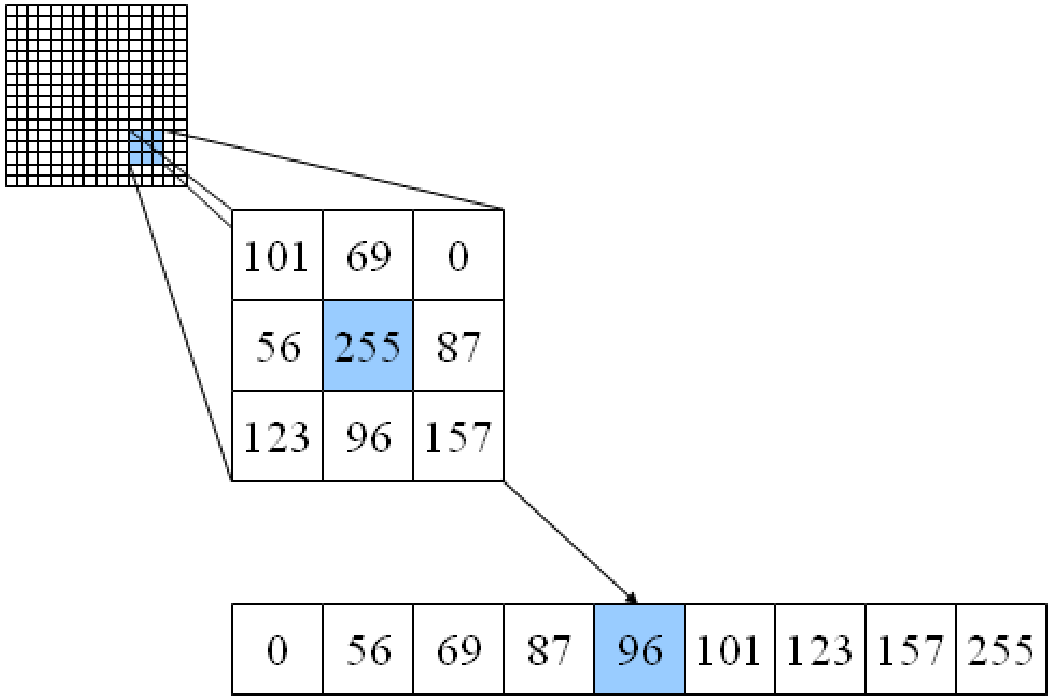
- Monotonic edges



courtesy: R. Fergus

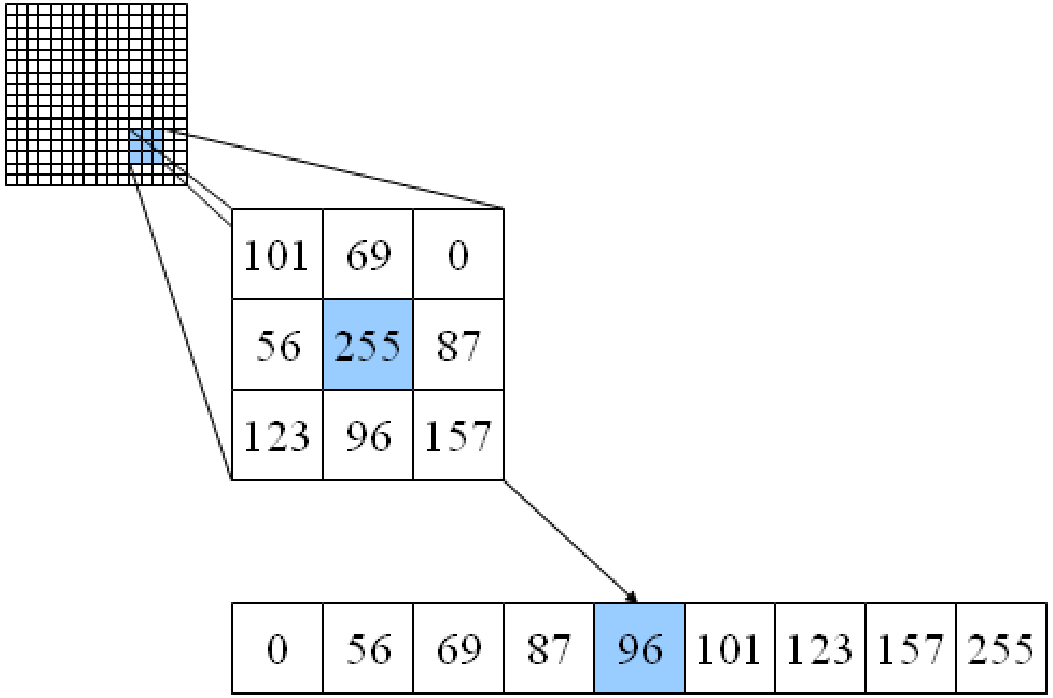
Median filter

- 2D
  - $N \rightarrow k \times k$ (e.g. $k = 3$)
  - k is window size



Median filter

- 2D
  - $N \rightarrow k \times k$ (e.g. $k = 3$)
  - k is window size

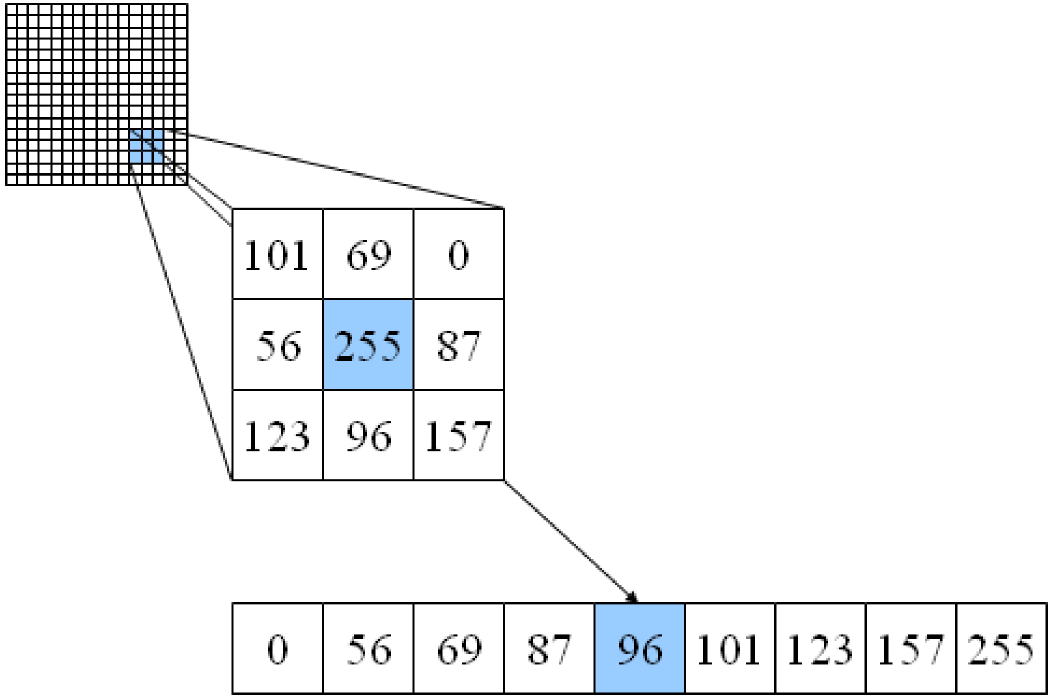



```
do
{
    while (a[i] < x) i++;
    while (a[j] > x) j--;
    int t = a[i];
    a[i] = a[j];
    a[j] = t;
} while (i < j);
```

Median filter

- 2D
  - $N \rightarrow k \times k$ (e.g. $k = 3$)
  - k is window size

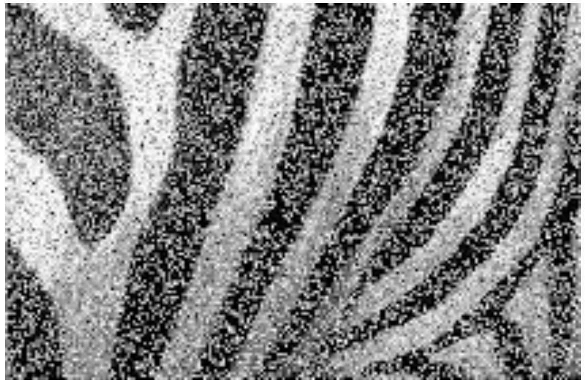
- partitioning around pivot
- do it till pivot is at N/2 location




```
do
{
    while (a[i] < x) i++;
    while (a[j] > x) j--;
    int t = a[i];
    a[i] = a[j];
    a[j] = t;
} while (i < j);
```

Median filter

- Effect of window size
  - best k depends upon image content, noise level & applications

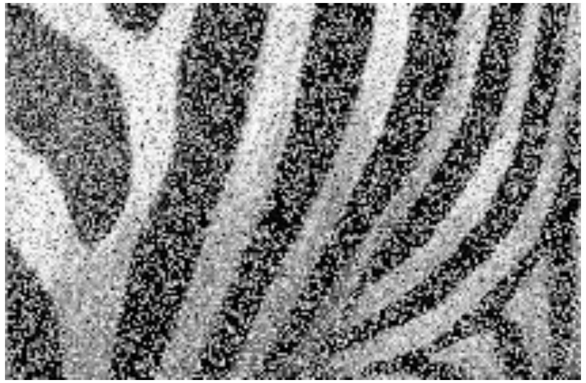


input

$k = 3$

Median filter

- Effect of window size
  - best k depends upon image content, noise level & applications



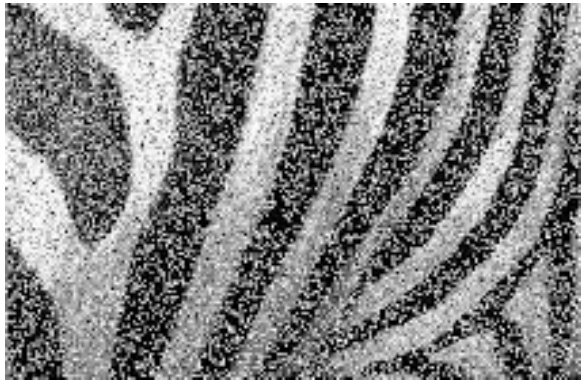
input

$k = 3$

$k = 5$

Median filter

- Effect of window size
  - best k depends upon image content, noise level & applications



input

$k = 3$

$k = 5$

$k = 7$

Median filter

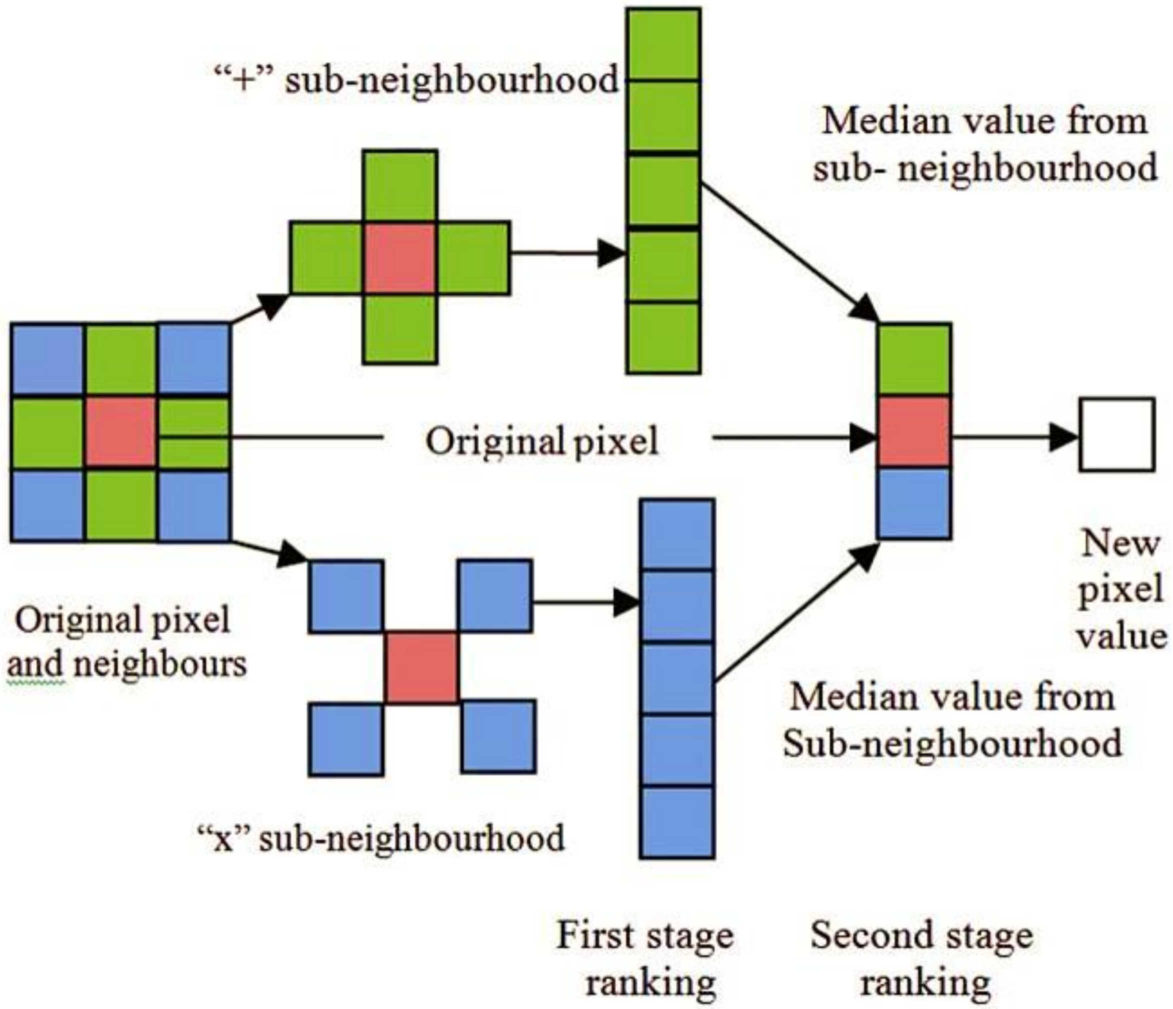
- Impulse noise

Median filter

- Impulse noise

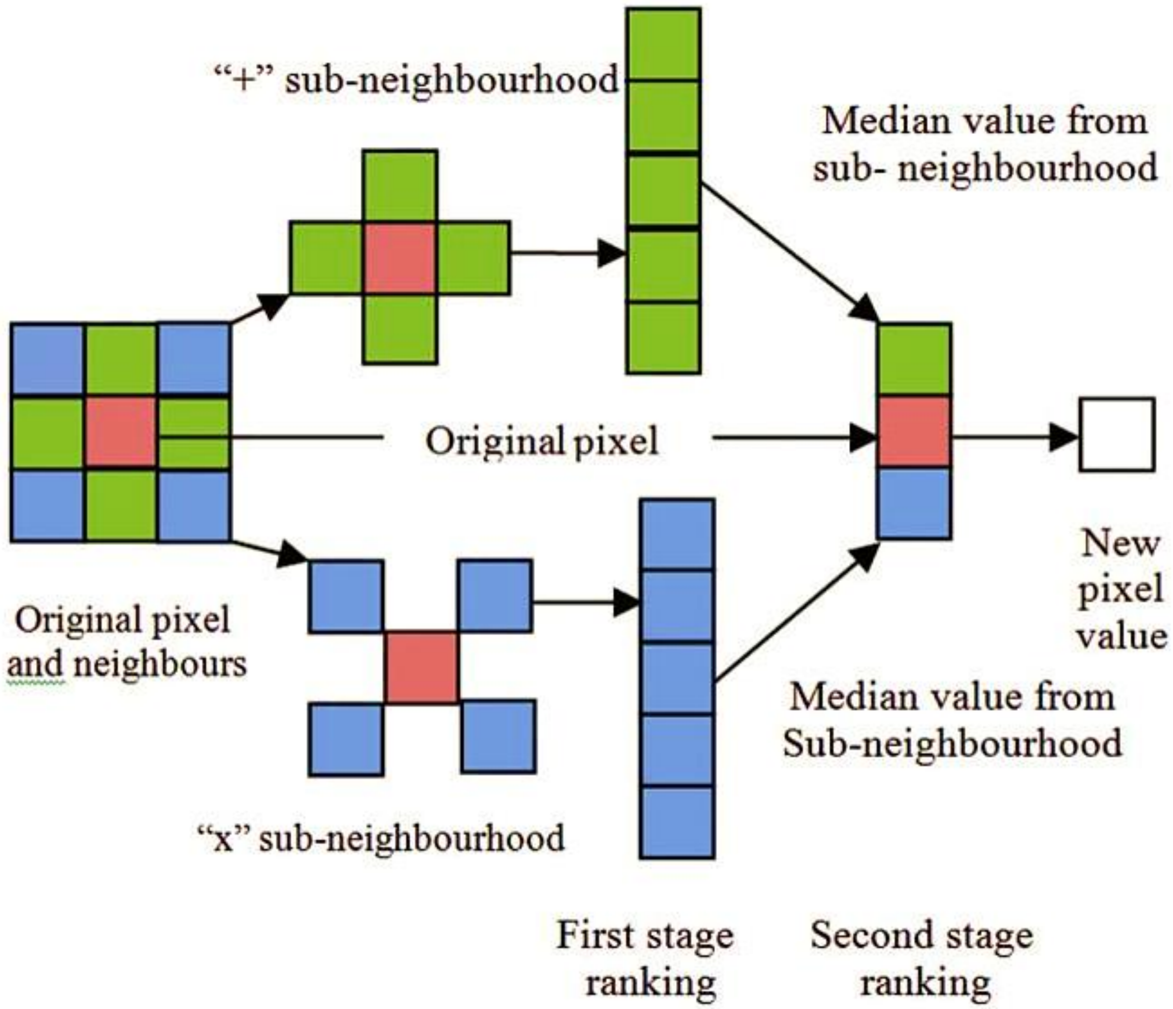
- Salt & pepper noise

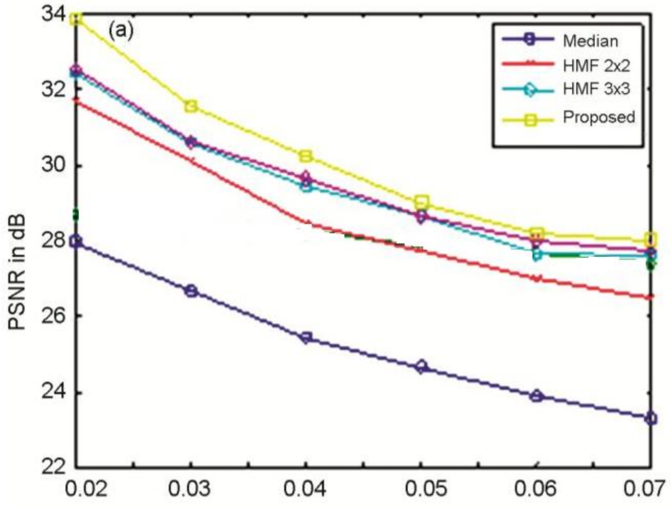
Hybrid median filter



courtesy: G. Umamaheswari

Hybrid median filter





courtesy: G. Umamaheswari

Burst mean filter

- Average across M images
  - fix the camera parameters

$$g(x, y, t_M) = \frac{1}{M} \sum_{k=0}^{M-1} f(x, y, t_k)$$

Burst mean filter

- Average across M images
  - fix the camera parameters

$$g(x, y, t_M) = \frac{1}{M} \sum_{k=0}^{M-1} f(x, y, t_k)$$

Burst mean filter

- Average across M images
  - fix the camera parameters

$$g(x, y, t_M) = \frac{1}{M} \sum_{k=0}^{M-1} f(x, y, t_k)$$

k=0 k=M-1

Burst mean filter

- Average across M images
  - fix the camera parameters

$$g(x, y, t_M) = \frac{1}{M} \sum_{k=0}^{M-1} f(x, y, t_k)$$

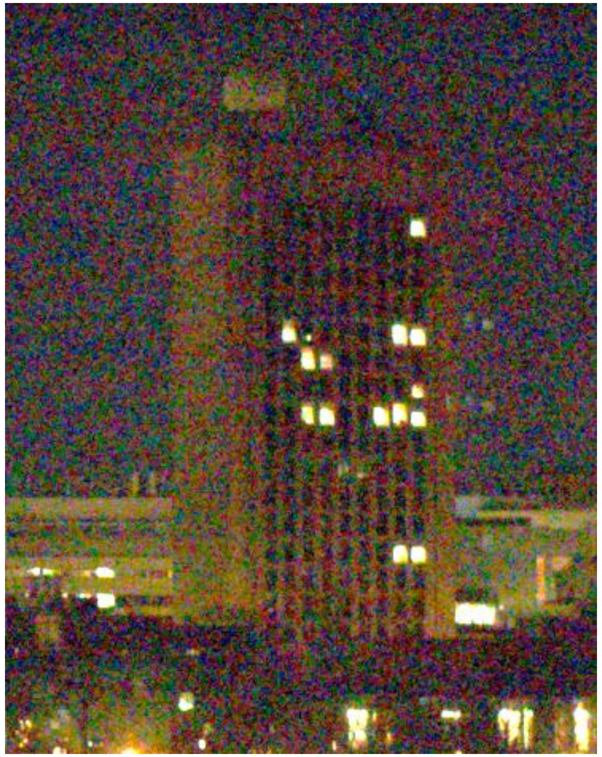
k=0 k=M-1

k=0 k=M-1

Burst mean filter

- Average across M images
  - fix the camera parameters

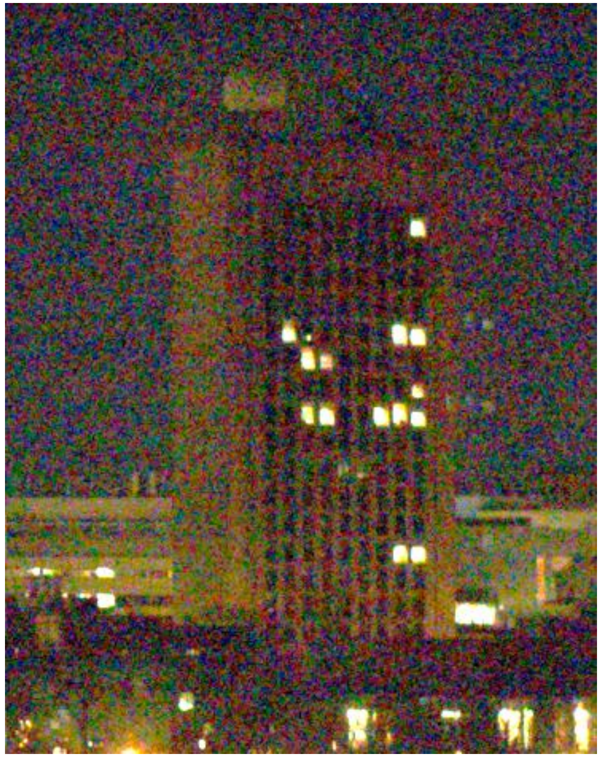
$M = 1$



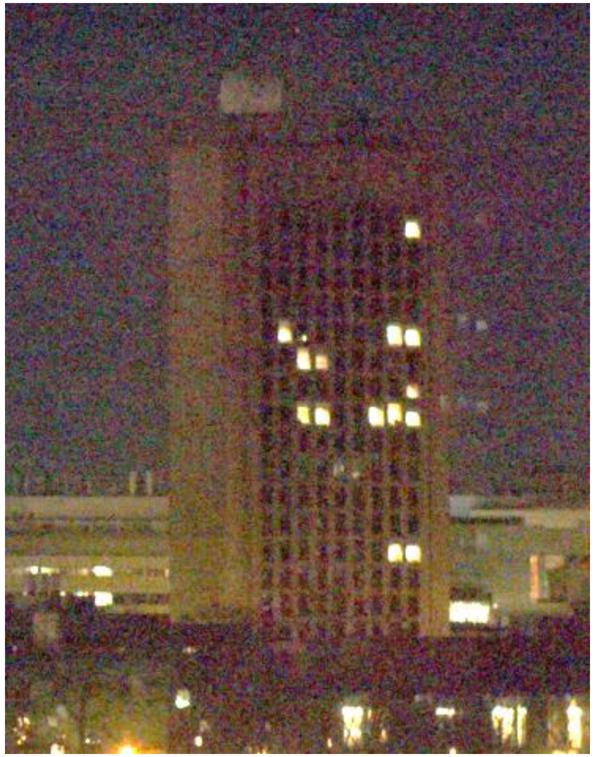
Burst mean filter

- Average across M images
  - fix the camera parameters

$M = 1$



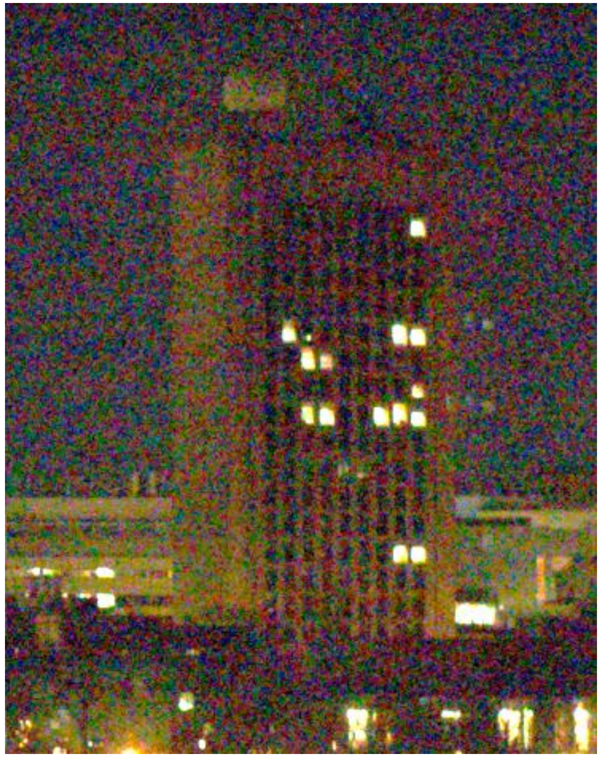
$M = 3$



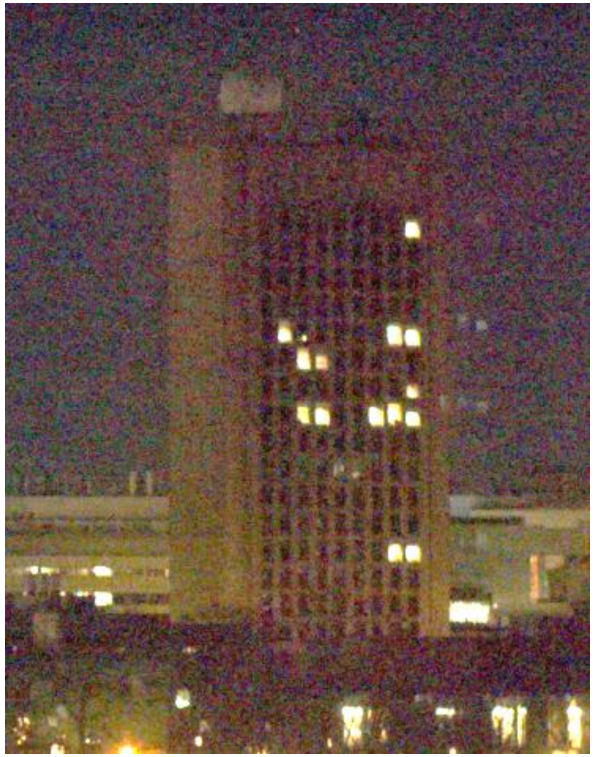
Burst mean filter

- Average across M images
  - fix the camera parameters

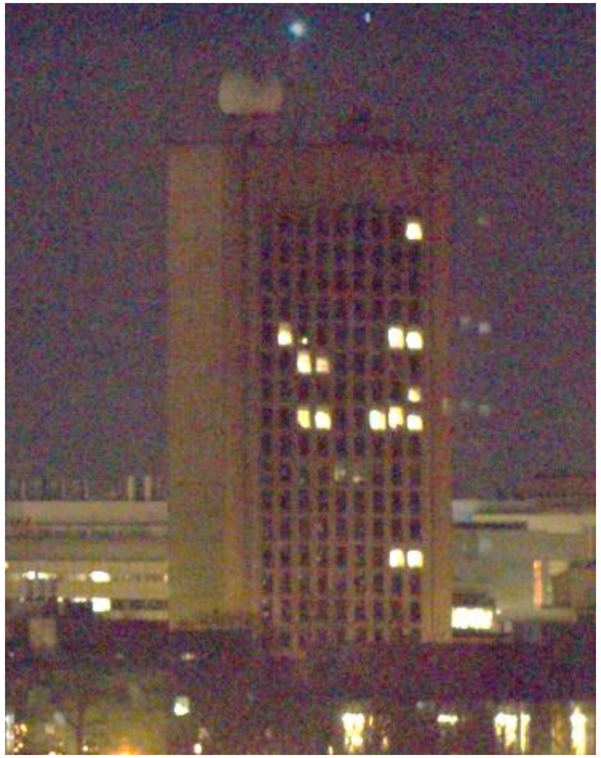
$M = 1$



$M = 3$



$M = 5$



Bilateral filter

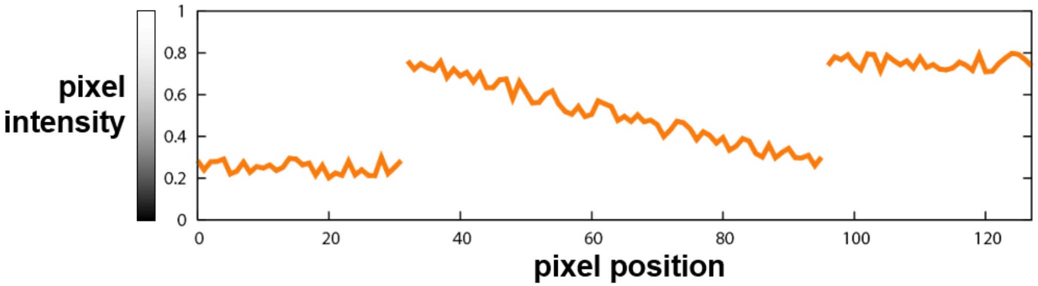
- Varying filter kernel
  - kernel depends upon image content

- 1D image

Bilateral filter

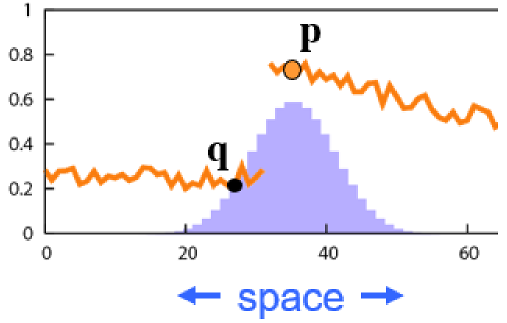
- Varying filter kernel
  - kernel depends upon image content

- 1D image



Bilateral filter

- Gaussian
  - kernel depends upon spatial dist

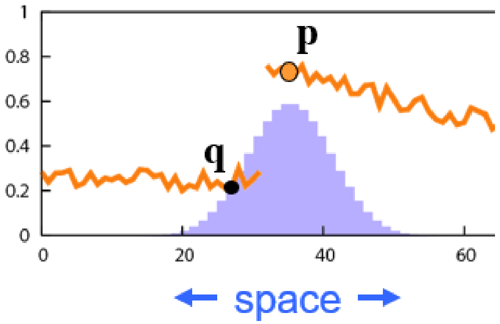


$$I^{\mathrm{b}}_{\mathbf{p}} = \sum_{\mathbf{q}\in\mathcal{S}} \underbrace{\boxed{G_{\sigma_{\mathrm{s}}}(\|\mathbf{p}-\mathbf{q}\|)}}_{\text{space}} I_{\mathbf{q}}$$

- Bilateral
  - kernel depends upon spatial + intensity range dist

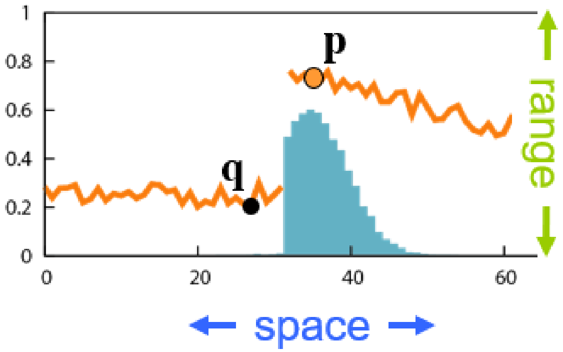
Bilateral filter

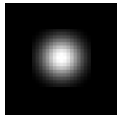
- Gaussian
  - kernel depends upon spatial dist

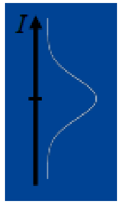


$$I^{\mathrm{b}}_{\mathbf{p}} = \sum_{\mathbf{q}\in\mathcal{S}} \underbrace{\boxed{G_{\sigma_{\mathrm{s}}}(\|\mathbf{p}-\mathbf{q}\|)}}_{\text{space}} I_{\mathbf{q}}$$

- Bilateral
  - kernel depends upon spatial + intensity range dist







$$I^{\mathrm{bf}}_{\mathbf{p}} = \underbrace{\boxed{\frac{1}{W^{\mathrm{bf}}_{\mathbf{p}}}}}_{\text{normalization}} \sum_{\mathbf{q}\in\mathcal{S}} \underbrace{\boxed{G_{\sigma_{\mathrm{s}}}(\|\mathbf{p}-\mathbf{q}\|)}}_{\text{space}} \underbrace{\boxed{G_{\sigma_{\mathrm{r}}}(|I_{\mathbf{p}}-I_{\mathbf{q}}|)}}_{\text{range}} I_{\mathbf{q}}$$

Bilateral filter

$$BF[I]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in \mathcal{S}} G_{\sigma_{\mathrm{s}}}(\|\mathbf{p} - \mathbf{q}\|)\, G_{\sigma_{\mathrm{r}}}(|I_{\mathbf{p}} - I_{\mathbf{q}}|)\, I_{\mathbf{q}},$$

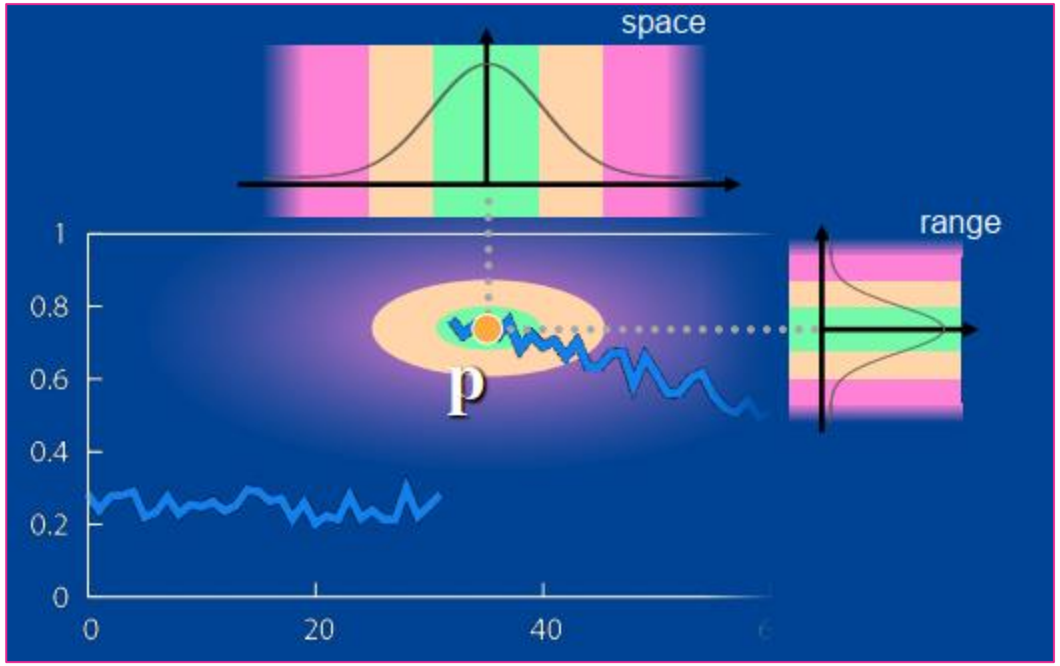
where

$$W_{\mathbf{p}} = \sum_{\mathbf{q} \in \mathcal{S}} G_{\sigma_{\mathrm{s}}}(\|\mathbf{p} - \mathbf{q}\|)\, G_{\sigma_{\mathrm{r}}}(|I_{\mathbf{p}} - I_{\mathbf{q}}|)$$

- BF
  - Amount of filtering is controlled via σ_s, σ_r
  - Spatial: σ_s – controls the influence of distant pixels
  - Range: σ_r – controls the influence of pixel intensity change

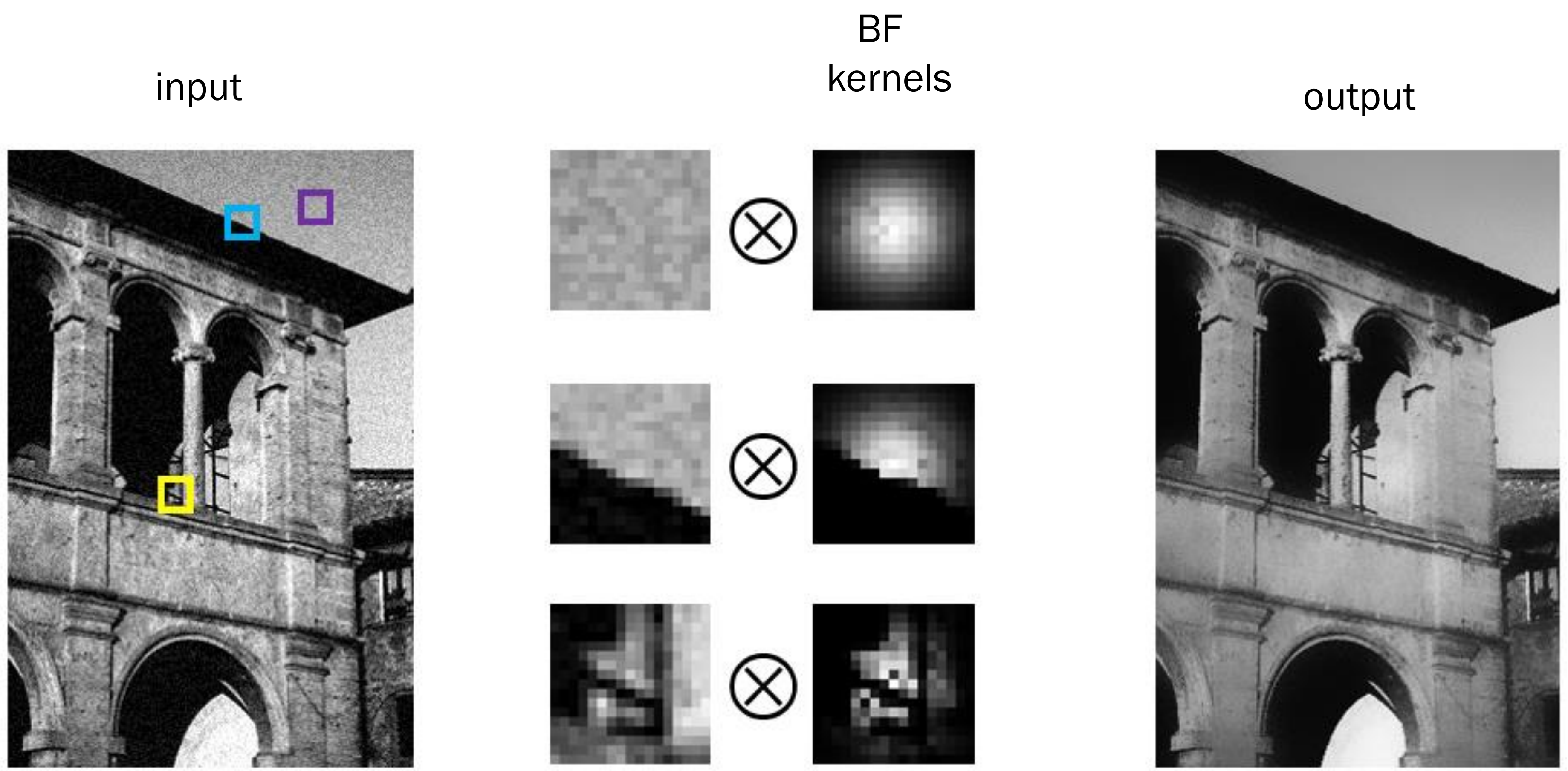
Bilateral filter

- Influence of pixels
  - pixels close in space as well as in range are the influencers, others are ignored



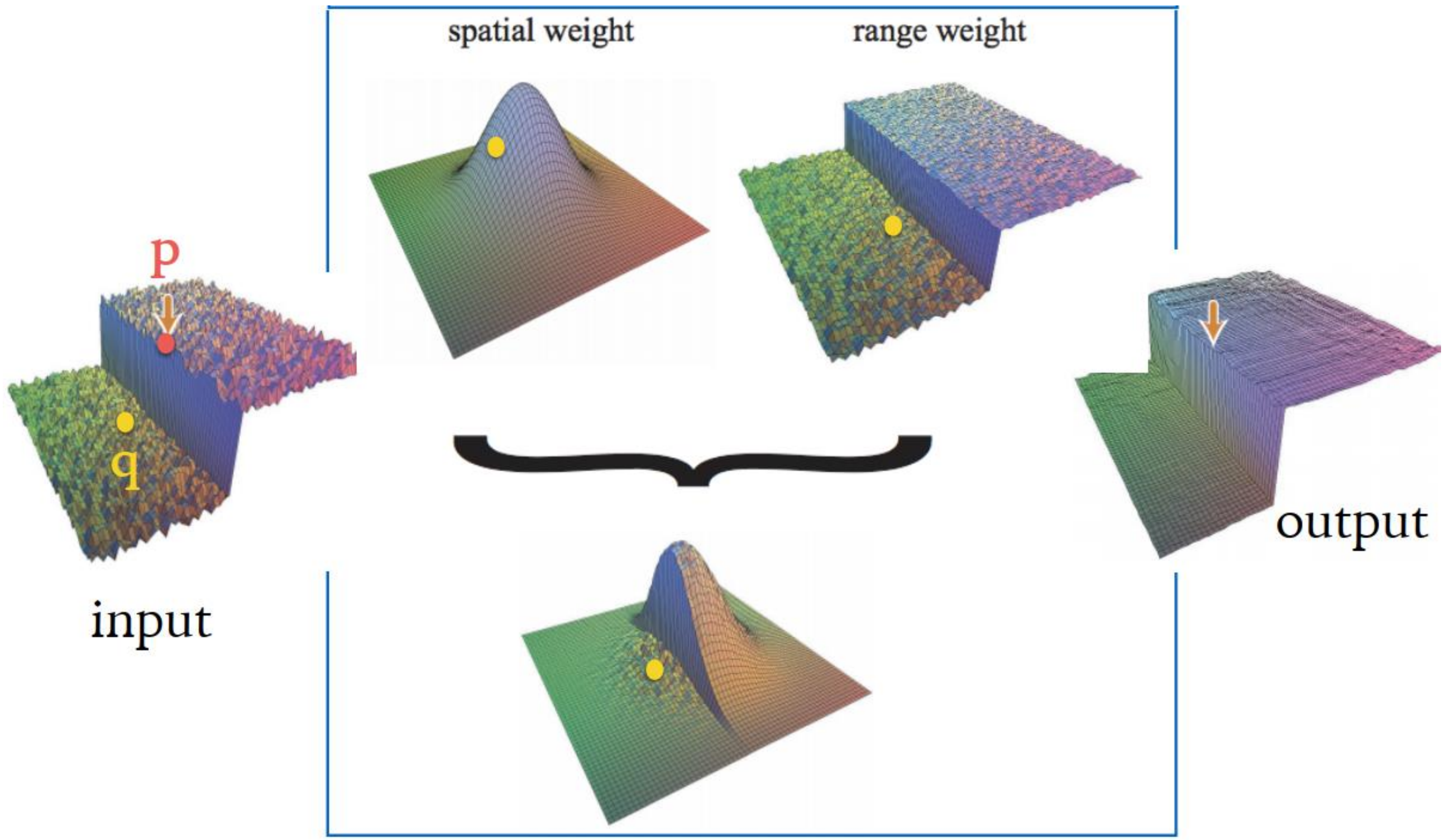
courtesy: MIT CSAIL

Bilateral filter



Bilateral filter

- Summary



courtesy: Paris et al.

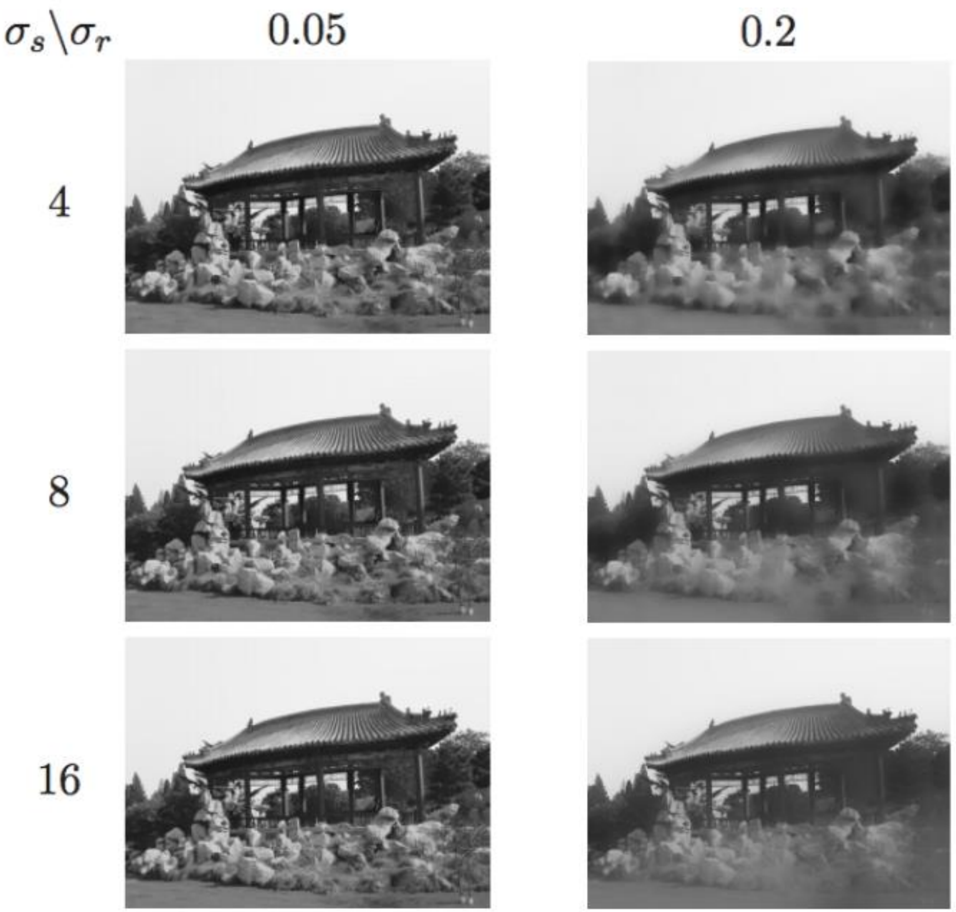
Bilateral filter

- Parameter effect

courtesy: Paris et al.

Bilateral filter

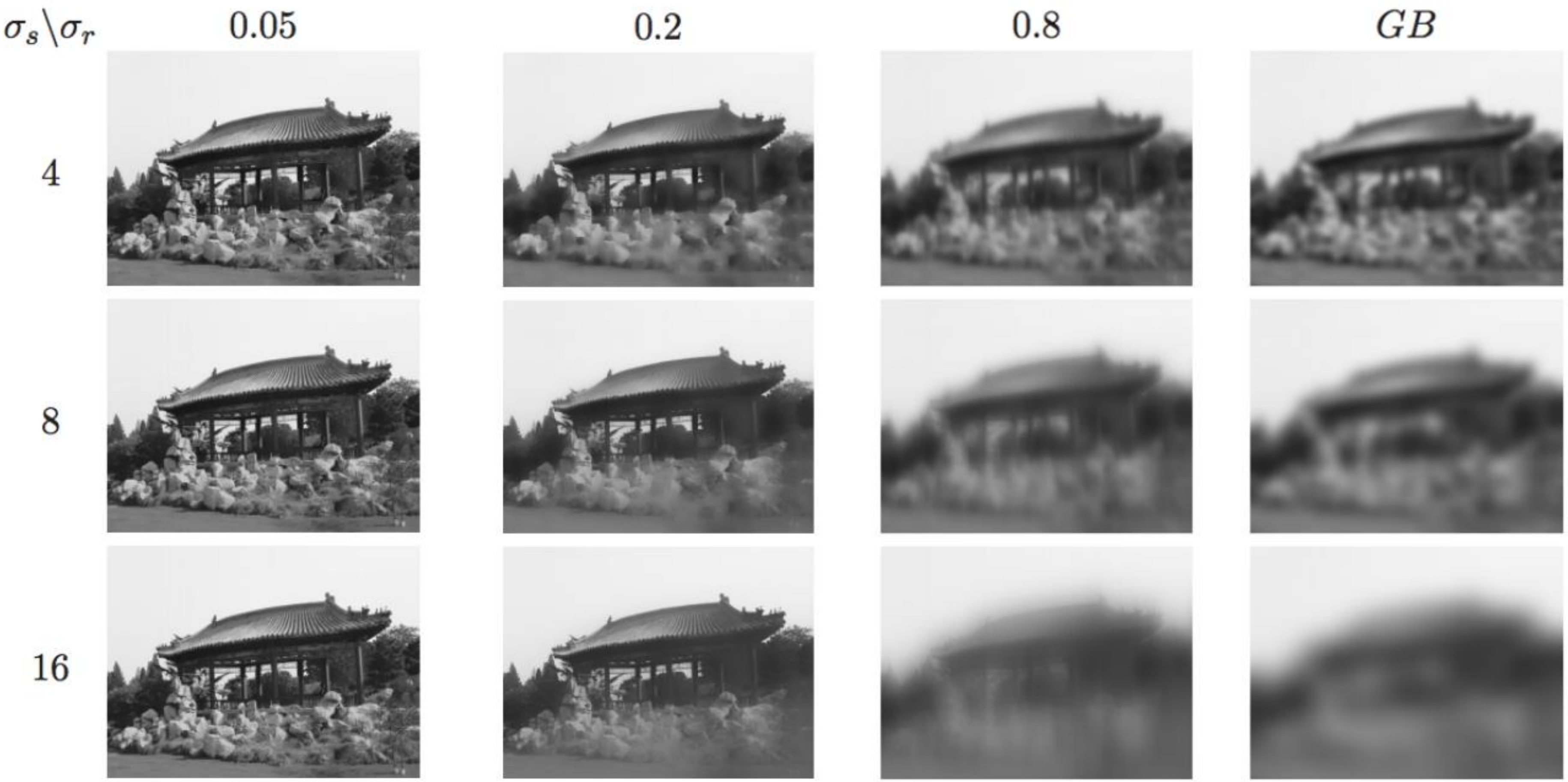
- Parameter effect



courtesy: Paris et al.

Bilateral filter

- Parameter effect



courtesy: Paris et al.

Bilateral filter

- Multiple iterations

input

BF iter-1

BF iter-5

Courtesy: C. Tomasi

Cartoon rendition

- $\sigma_s \uparrow$ & iterate

References

- Denoising by filtering

References

- Denoising by filtering

- CF Wstin Ron Kikiniis, H Knutsson, “Handbook of medical imaging”, 2000
- J Joseph, BN Anoop, J Williams, “A modified unsharp masking with adaptive threshold…”, Multimedia tools and applications, 2018
- Y Zhu and C Huang, “An improved median filtering algorithm for image noise reduction”, SSDMS Procedia 2012
- TS Huang, GJ Yang, GY Tang, “A fast two-dimensional median filtering algorithm”, IEEE Tran ASSP, 1979
- C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color imaegs”, IEEE ICCV 1998
- S. Paris and F. Durand, “A fast approximation of the bilateral filter using a signal processing approach”, ECCV 2006
- Q. Yang, KH Tan, N. Ahuja, “Real time O(1) bilateral filtering”, IEEE CVPR, 2009
- H Winnemoller, SC Olsen, B Gooch, (Cartoon rendition) “Real time video abstraction”, SIGGRAPH, 2006