“Fixing the Gaussian Blur”: the Bilateral Filter

Sylvain Paris – MIT CSAIL
Blur Comes from Averaging across Edges

Same Gaussian kernel everywhere.
Bilateral Filter [Aurich 95, Smith 97, Tomasi 98]

No Averaging across Edges

The kernel shape depends on the image content.
Bilateral Filter Definition: an Additional Edge Term

Same idea: weighted average of pixels.

\[
BF [I]_p = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(\|I_p - I_q\|) I_q
\]

- New
- Not new
- normalization factor
- space weight
- range weight
Illustration a 1D Image

- 1D image = line of pixels
  
  ![1D Image Illustration]

- Better visualized as a plot

![Plot of 1D Image]
Gaussian Blur and Bilateral Filter

Gaussian blur

$$GB[I]_p = \sum_{q \in S} G_{\sigma} (\| p - q \|) I_q$$

space

Bilateral filter

[Aurich 95, Smith 97, Tomasi 98]

$$BF[I]_p = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s} (\| p - q \|) G_{\sigma_r} (\| I_p - I_q \|) I_q$$

range

Normalization
Bilateral Filter on a Height Field

\[ BF \left[ I \right]_p = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s} \left( \| p - q \| \right) G_{\sigma_r} \left( \| I_p - I_q \| \right) I_q \]
Space and Range Parameters

\[
BF[I]_p = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(\|I_p - I_q\|) I_q
\]

- space \( \sigma_s \): spatial extent of the kernel, size of the considered neighborhood.
- range \( \sigma_r \): “minimum” amplitude of an edge
Influence of Pixels

Only pixels close in space and in range are considered.
Exploring the Parameter Space

input

\( \sigma_s = 2 \)

\( \sigma_s = 6 \)

\( \sigma_s = 18 \)

\( \sigma_r = 0.1 \)

\( \sigma_r = 0.25 \)

\( \sigma_r = \infty \)

(Gaussian blur)
Varying the Range Parameter

\( \sigma_r = 0.1 \quad \sigma_r = 0.25 \quad \sigma_r = \infty \) (Gaussian blur)

\( \sigma_s = 2 \)

\( \sigma_s = 6 \)

\( \sigma_s = 18 \)
$\sigma_r = 0.1$
$\sigma_r = 0.25$
\[ \sigma_r = \infty \]
(Gaussian blur)
Varying the Space Parameter

\[ \sigma_s = 2 \]
\[ \sigma_s = 6 \]
\[ \sigma_s = 18 \]

\[ \sigma_r = 0.1 \]
\[ \sigma_r = 0.25 \]
\[ \sigma_r = \infty \] (Gaussian blur)
\[ \sigma_s = 2 \]
\[ \sigma_s = 6 \]
How to Set the Parameters

Depends on the application. For instance:

- **space parameter:** proportional to image size
  - e.g., 2% of image diagonal

- **range parameter:** proportional to edge amplitude
  - e.g., mean or median of image gradients

- independent of resolution and exposure
A Few More Advanced Remarks
Bilateral Filter Crosses Thin Lines

- Bilateral filter averages across features thinner than $\sim 2\sigma_s$
- Desirable for smoothing: more pixels = more robust
- Different from diffusion that stops at thin lines
Iterating the Bilateral Filter

\[ I_{(n+1)} = BF[I_{(n)}] \]

- Generate more piecewise-flat images
- Often not needed in computational photo.
1 iteration
2 iterations
4 iterations
Bilateral Filtering Color Images

For gray-level images

\[
BF[I]_p = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(\|I_p - I_q\|) I_q
\]

intensity difference

For color images

\[
BF[I]_p = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(\|C_p - C_q\|) C_q
\]

color difference

The bilateral filter is extremely easy to adapt to your need.
**Hard to Compute**

- **Nonlinear**
  \[
  BF[I]_p = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(|I_p - I_q|) I_q
  \]

- **Complex, spatially varying kernels**
  - Cannot be precomputed, no FFT...

- **Brute-force implementation is slow > 10min**
Questions?