

Bilateral Filter

Aydın, Anıl Can

November 30, 2016

This document is constructed to clarify the steps of the implementations of Bilateral Filter on this repository. There are two implementations with same logic on Python and C++. They are constructed in the context of the classes of Image Understanding.

A bilateral filter is a non-linear, edge-preserving and noise-reducing smoothing filter for images. The intensity value at each pixel in an image is replaced by 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 by systematically looping through each pixel and adjusting weights to the adjacent pixels accordingly.

The bilateral filter is defined as:

$$I^{filtered}(x) = \frac{1}{W_p} \sum_{x_i \in \Omega} I(x_i) f_r(||I(x_i) - I(x)||) g_s(||x_i - x||),$$

where the normalization term

$$W_p = \sum_{x_i \in \Omega} f_r(||I(x_i) - I(x)||) g_s(||x_i - x||)$$

ensures that the filter preserves image energy and

- $I^{filtered}$ is the filtered image;
- I is the original input image to be filtered;
- x are the coordinates of the current pixel to be filtered;
- Ω is the window centered in x ;
- f_r is the range kernel for smoothing differences in intensities. This function can be a Gaussian function;

- g_s is the spatial kernel for smoothing differences in coordinates. This function can be a Gaussian function;

As mentioned above, the weight W_p is assigned using the spatial closeness and the intensity difference. Consider a pixel located at (i, j) which needs to be denoised in image using its neighbouring pixels and one of its neighbouring pixels is located at (k, l) . Then, the weight assigned for pixel (k, l) to denoise the pixel (i, j) is given by:

$$w(i, j, k, l) = e^{(-\frac{(i-k)^2+(j-l)^2}{2\sigma_d^2} - \frac{\|I(i, j) - I(k, l)\|^2}{2\sigma_r^2})}$$

where σ_d and σ_r are smoothing parameters and $I(i, j)$ and $I(k, l)$ are the intensity of pixels (i, j) and (k, l) respectively. After calculating the weights, normalize them.

$$I_D(i, j) = \frac{\sum_{k,l} I(k, l) * w(i, j, k, l)}{\sum_{k,l} w(i, j, k, l)}$$

where I_D is the denoised intensity of pixel (i, j) .

References

- [1] https://en.wikipedia.org/wiki/Bilateral_filter