Image Processing Assignment: Bilateral and Joint Bilateral Filters

November 2018

A Bilateral Filters

A.1

The bilateral filter is a method of smoothing by spacial filtering that attempts to preserve edges.This is done by taking both the spatial proximity and similarity in intensity into account. The most common way of doing this is by using two Gaussian functions, one that takes the physical distance between two pixels as an input, and a second that takes the absolute difference in intensity as its input. The Gaussian function gives values according to a normal distribution, which has probability density function:

$$g(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

For mean μ and variance σ^2. Since it is optimal to attribute the largest values to the closest pixels, μ is taken to be zero, giving:

$$g(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-x^2}{2\sigma^2}}$$

This weighted value is then multiplied by the initial value of the input pixel and summed over a neighbourhood. It is then normalised by dividing by the sum of the weighted values. For a neighbourhood around $I_{in}(i,j)$ denoted by Ω, this gives:

$$I_{out}(i,j) = \frac{\sum_{p\in\Omega} g_{\sigma_s}(\mid p-(i,j)\mid) g_{\sigma_r}(\mid I_p - I_{in}(i,j)\mid) I_p}{\sum_{p\in\Omega} g_{\sigma_s}(\mid p-(i,j)\mid) g_{\sigma_r}(\mid I_p - I_{in}(i,j)\mid)}$$

Where σ_r and σ_s are the standard deviations ($\sqrt{\sigma^2}$) of the Gaussian functions. Changing these parameters affects how wide the distribution of values is, effectively giving more/less weight to pixels further away, or more different, from the input.

Because the two values of the Gaussian function are multiplied together, smoothing only occurs when pixels are both somewhat close and similar, since if one of the values is close to zero, the entire term becomes negligible. This means that pixels that are very similar, but far away, or close but very different, have little effect on the output.

A bilateral filter can be iterated: applying the filter multiple times, each time to the result of the previous iteration. Doing this over a large number of iterations produces an image that approaches being piecewise constant, appearing heavily stylised, or cartoonish.

A.2

See 'Bilateral Filter.py' for implementation of the bilateral filter using OpenCV.

A.3

B Joint Bilateral Filters

B.1

B.2

See 'Joint Bilateral Filter.py' for implementation of the joint bilateral filter using OpenCV.

B.3