Bilateral filters

The purpose of a filter is to remove noise out of an image whilst retaining the information. A bilateral filter is a type of noise reducing filter for images that is non-linear, meaning each output pixel is not a linear function of the corresponding input pixel's neighbourhood. It also uses an edge preserving smoothing technique so sharp edges are preserved when the image is smoothed.

The bilateral filtered image is defined by:

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

where W_p is a normalisation factor.

$$W_p = \sum_{x_i \in \Omega} G_{\sigma_r}(\|I(x_i) - I(x)\|)\, G_{\sigma_s}(\|x_i - x\|)$$

I represents the original image to be filtered and each x represents the coordinates of the current pixel. Ω is the pixel neighbourhood centred in x. G_{σ_r} is the range (Gaussian) kernel that decreases the influence of pixels x_i with intensity values that differ from $I(x)$ and G_{σ_s} is the spatial (Gaussian) kernel that decreases the influence of pixels based on differences in coordinates. The parameters σ_r and σ_s measures the amount image I gets filtered. Increasing σ_s tends to smooth larger features and by raising σ_r, the similarity to Gaussian blur becomes increasingly evident as the range Gaussian progressively gets flatter.

As bilateral filtering multiplies space and range weights together, no smoothing occurs as soon as one of the weights get close to zero.

test 1 output

test 2 output

Bilateral filter implementation

The implemented OpenCV bilateral filter uses the following parameters:

src: the input image (I)
d: the diameter of each pixel neighbourhood (Ω) that is used during filtering.

sigmaColour: the amount the image gets filtered in the colour space (σ_r).
sigmaSpace: the amount the image gets filtered in the coordinate space (σ_s).

The number of pixels processed in a neighbourhood is directly proportional to the diameter 'd', meaning larger values of 'd' increases the amount of noise filtering however heavily impacts the time performance. By using single digit figures for the diameter in the test cases, the processing time is very reasonable.

By increasing the 'sigmaColour' parameter, the greater the range of acceptable colours within the pixel neighbourhood becomes. This allows more colours to be blended together, resulting into larger areas of image having similar colours. Comparatively, the larger value of 'sigmaSpace', the greater the influence of farther pixels in the neighbourhood have on each other.

Setting 'sigmaColour' and 'sigmaSpace' as the same value in both cases removes biasness towards either Gaussian calculation. To ensure to keep the sharpness of 'test 1', low sigma values of thirty were chosen. Decreasing the value further led to un-noticeable changes to the image. Starting with sigma values of over one hundred for 'test 2' led to the face looking unrealistic. By lowering the sigma values to sixty enhanced defining features of the face whilst retaining the smoothness of the image.

Joint bilateral filters

A joint bilateral filter is a variant of the bilateral filter that takes two photos, one with and one without the flash and combines them together. To avoid motion of either the photographer or scene, the two photos must be taken in quick succession. Fundamentally, an image is composed of high and low frequencies, where high frequency components signify edges and low frequency components signify smooth areas. Typically, flash images contain high frequency flash detail whereas non-flash images have ambient qualities. Joint bilateral filters noise-reduce images by smoothening the colour space of the low-frequency ambient image whilst preserving the edges emphasised in the flash image. Often low-light environments have warm atmospheres but suffer from noise and blur. Alternatively, flash photography sharpens images but lies susceptible to redeyes and harsh lighting. The aim of creating a joint bilateral filtered image is to combine the best features of the two different lightings into one image, and is defined by:

$$I^{filtered}(x) = \frac{1}{W_p} \sum_{x_i \in \Omega} G_{\sigma_r}(\|F(x_i) - F(x)\|)\, G_{\sigma_s}(\|x_i - x\|) I(x_i)$$

where W_p is a normalisation factor.

$$W_p = \sum_{x_i \in \Omega} G_{\sigma_r}(\|F(x_i) - F(x)\|)\, G_{\sigma_s}(\|x_i - x\|)$$

The joint bilateral filter shares the same variables ($I, x, \Omega, G_{\sigma_r}$ and G_{σ_s}) as the bilateral filter, and only differs by the condition F, representing the flash image. Replacing the ambient image with a flash image when computing G_{σ_r} preserves edges that may not be evident in the ambient image. This is because illuminating an environment increases the signal-to-noise ratio (SNR), providing a better estimate of high-frequency details.

Joint bilateral filter implementation

The implemented joint bilateral filter takes in the following parameters:

no_flash_image:	the ambient image.
flash_image:	the image used when performing the Gaussian function.
new_image:	the output image that is being updated by the function.
x, y:	the coordinates of the central pixel (x) in the kernel.
d:	the diameter of each pixel neighbourhood (Ω) that is used during filtering.
sigma_r:	the amount the image gets filtered in the colour space (σ_r).
sigma_s:	the amount the image gets filtered in the coordinate space (σ_s).

As mentioned in A3, the number of pixels processed in a neighbourhood is directly proportional to the diameter 'd'. Likewise, this applies for the joint bilateral filter, and hence the diameter has been set to 5 pixels to ensure the processing time is reasonable.

Looking at the flash image, there is little noise. Thus, sigma_r can be set to a very small value to ensure that the Gaussian kernel for colour range assigns weights for neighbouring pixels that will not over or under-blur the non-flash image. Reading "Digital photography with flash and no-flash image pairs" suggests 0.1% of the total range of colour values as a good approximation for sigma_r. For the specific image, having calculated the range of colours to be 236, I set sigma_r to 0.236.

After trialling multiple values for sigma_s, I settled on the value 5. Larger values for sigma_s tended to overpower sigma_r, leaving the filtered image to look far too smooth. Setting sigma_s with values less than one has minimal effect on the ambient image, causing the filtered image to look very similar to the original ambient image.