Homomorphic Processing

Consider an image with bright sunlight: there are areas where there is a lot of lighting, and there are areas in the shadow, where there is only very little light. This means we have a very high dynamic range of light intensities. The eye can process a very high dynamic range, because it has a non-linear (log) pre-processing, and also something like an adaptive gain control.

But common displays don’t have such a high dynamic range, so that they cannot display such images faithfully. Similar, common cameras don’t have a high dynamic range as the eye. But the main limiting factor are the displays.

Assume, we have an image with a fairly high dynamic intensity range, and we would like to display it on a display with a more limited dynamic range, such that we can still see details in the dark and the bright areas. How can we do it?

A possible approach would be to apply a non-linear display function, a suitable Gamma, as described above. This would increase the lower intensities on the display, to make them more visible.

But this has limits with very high dynamic ranges, the contrasts for the dark and bright parts of an image might not become high enough using just the Gamma method.

What else could we do? We can take a look at the image formation process. A simple model is to separate the lighting effects (the illumination, $i(n_1, n_2)$) from the reflection effects...
\( r(n_1, n_2) \). The latter show the actual information about an object. The resulting image \( f(n_1, n_2) \) is:

\[
f(n_1, n_2) = i(n_1, n_2) r(n_1, n_2)
\]

We can now assume that the illumination \( i(n_1, n_2) \) only varies slowly over an image, which means we get mostly low spatial frequencies for it, whereas the reflectance \( r \) contains the fine details of the image, and hence has stronger high frequency components. To display this image on our low dynamic range display, the approach now is to reduce the effect of the illumination \( i \) and increase the effect of the reflectance \( r \). This would bring the resulting image closer to a diffuse illumination.

To do this, we first need to separate these two factors. To apply linear systems for the enhancement of the reflectance \( r \), we first need to turn the multiplication of \( i \) and \( r \) into a sum. We obtain this by applying the Logarithm:

\[
\log f(n_1, n_2) = \log i(n_1, n_2) + \log r(n_1, n_2)
\]

Now we can separate the reflectance \( r \) from the illumination \( i \) by linear filtering. We apply low pass filtering to obtain the illuminance component, and high pass filtering to obtain the reflectance component. Then we can apply different factors to these two components. We apply a factor bigger than 1 for the reflectance, and a factor smaller than 1 for the illuminance. After that, we can add those modified two components again, and apply an exponential function to get
our enhanced image. The block diagram of this processing can be seen in the following picture:

![Block Diagram](image)

Observe: This can also be seen as unsharp masking in the logarithmic domain.

An example of this processing can be seen in following picture:
Observe: The effect of this process is an unequal treatment of different areas of the image, unlike for instance the Gamma processing, which applies the same function to all areas of an image. The effect here is to apply different functions to dark areas than to bright areas in an image, which can be seen as an adaptive processing.

**Noise Smoothing**

Assume we have a noisy image, for instance from camera noise. How can we reduce the noise?
Noise is primarily wideband in character, unlike images which are mostly low pass in character. To reduce the total noise power, we can hence apply a low pass filter to the image, since the high frequencies of the noisy image contain mostly noise.

The disadvantage of this method is, that also the high frequency components of the image are lost, which might not have much power in the signal, but which are still important to maintain sharp edges in the image. Hence the low pass filtered version looks blurred, more unsharp.

A different approach is the so-called Median Filtering. This is especially efficient for the so-called Salt and Pepper noise, which consist mainly of outliers in our picture, which are a few pixels which have a big error (for instance from bit errors).

Definition of Median: this is the value of a sequence of numbers, where half of the values are larger than the median, and half of the values are smaller than the median. (Unlike the average, which is the sum divided by the number of samples).

Example: Take a sequence of values: 1,3,4,6,7. Here, 4 is the median, which is independent of the highest or lowest value in the sequence (outliers), unlike the average. The average is 21/5=4.2.

The median filter looks at a window over the image, and replaces the value in the center of the window with the
median over that window. This window is the shifted, sample by sample, over the entire image. Crucial here is the size of the window, for the strength of the median filtering. This has the advantage that it maintains edges, it is still sharp. Examples of the effect:

(Pictures from: Jae. S. Lim, Two-Dimensional Signal and Image Processing)
Figure 8.18 Illustration of a median filter’s capability to remove impulsive values. (a) One-dimensional sequence with two consecutive samples significantly different from surrounding samples; (b) result of lowpass filtering the sequence in (a) with a 5-point rectangular impulse response; (c) result of applying a 5-point median filter to the sequence in (a).