0.1 Image Enhancement

- *Algorithm:* Contrast Enhancement (algo. 1)
- *Input:* An image with $m \times n$ pixels.
- *Complexity:* $\mathcal{O}(mn)$
- *Data structure compatibility:* N/A
- *Common applications:* Make images which are overexposed or underexposed better to recognize, or enhance images with too dark or too bright background.

Problem. Image Enhancement

Sometimes an image would be too hard for people to recognize some important details in it due to overexposure, underexposure, or background light problems. Therefore, how to enhance the image, such that the details can be highlighted and clear to see would be the problem.

Description

There are many ways to realize image enhancement. This time I would focus on the contrast enhancement method which takes use of grey-level histogram.

In image editing and manipulation, contrast enhancement is the method to adjust the histogram of an image to shift pixels between the lightest and darkest parts of the image [1].

Usually, contrast enhancement is performed in two steps, firstly a tonal enhancement and then the contrast stretch. Tonal enhancements "improve the brightness differences in the dark", grey and bright regions, while a contrast stretch increases the brightness differences "uniformly across the dynamic range of the image. [2]"

Grey-level(GL) Histogram

Once given an image, a corresponding grey-level histogram will be created by counting how many times each grey-level value (from 0 to 255) are present in the image. For example, Fig. 1 shows the grey-level histogram of an image.

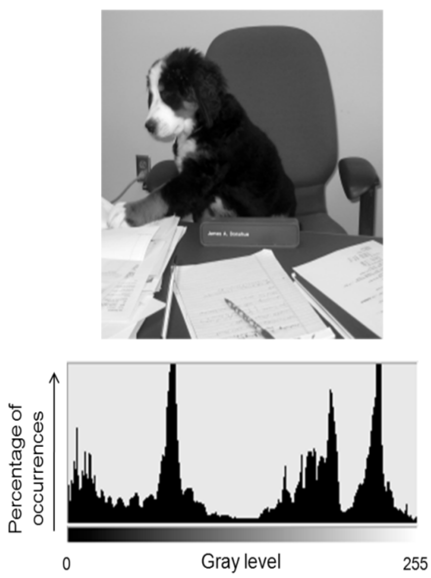

Figure 1: The grey-level histogram of an image

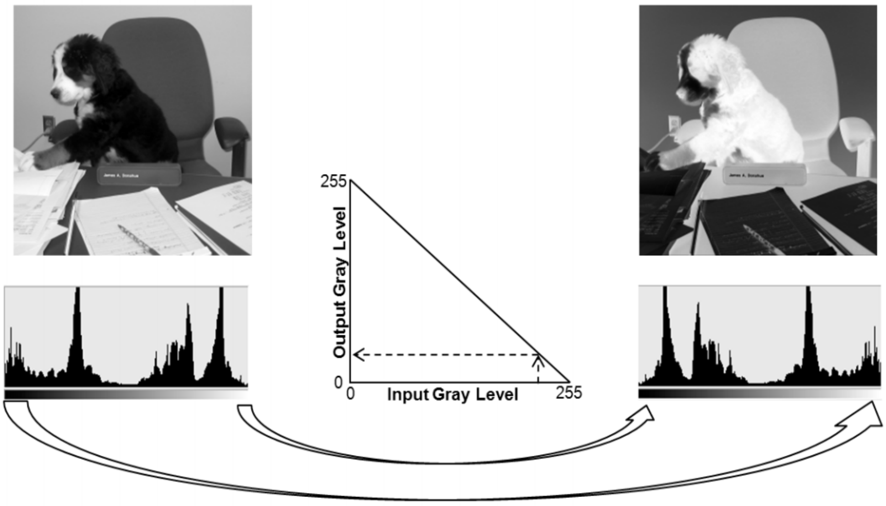

Figure 2: Contrast Reversal

After getting the grey-level histogram, a mapping function would be applied to adjust the current grey-level GL to a new one GL' to realize enhancement. The simplest method is directly contrast reversal, which means that for every pixel in the image, the grey-level value $g(x, y)$ ((x, y) denotes the location of the pixel) would be mapped into $g'(x, y) = 255 - g(x, y)$, the effect of which can be illustrated by Fig. 2.

Contrast Stretch

A high-contrast easy-to-recognize image contains gray-level values of the full range 0 to 255. One method is to create a remapping such that the lowest GL_{min} will be mapped to 0 and the highest GL_{max} will be remapped to 255. Also, GL between them would be linearly remapped between 0 and 255, with the transformation equation as

$$g'(x, y) = \lfloor \frac{255}{GL_{max} - GL_{min}} (g(x, y) - GL_{min}) \rfloor$$

If we want to customize the remapped GL'_{min} and GL'_{max}, then we can create the following generalized linear transformation.

$$g'(x, y) = \lfloor \frac{GL'_{max} - GL'_{min}}{GL_{max} - GL_{min}} (g(x, y) - GL_{min}) + GL'_{min} \rfloor$$

In this way, the relative shape of the histogram would be unchanged, but the range would be expanded to fill all between $[GL'_{min}, GL'_{max}]$. The transformation function is shown in Fig. 3. And the comparison between the original image and the image after stretch is depicted in Fig. 4.

Tonal Enhancement

The contrast stretch method only deal with linear transformation of grey-level values of an image. Sometimes we may need nonlinear transformations such that we can enhance some grey-level regions while potentially reduce the contrast in other regions [2].

The most common nonlinear transformation is the gamma correction [3], which uses an exponent of $1/\gamma$ to preprocess the image to produce a linear response to brightness. The transformation formula is as follows

$$g'(x, y) = \lfloor 255 \cdot (\frac{g(x, y)}{255})^{\frac{1}{\gamma}}) \rfloor$$

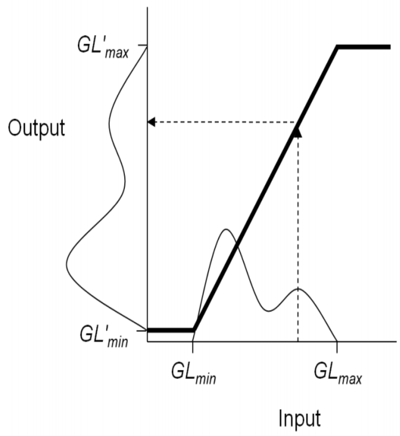

Figure 3: Linear Transformation of the grey-level value

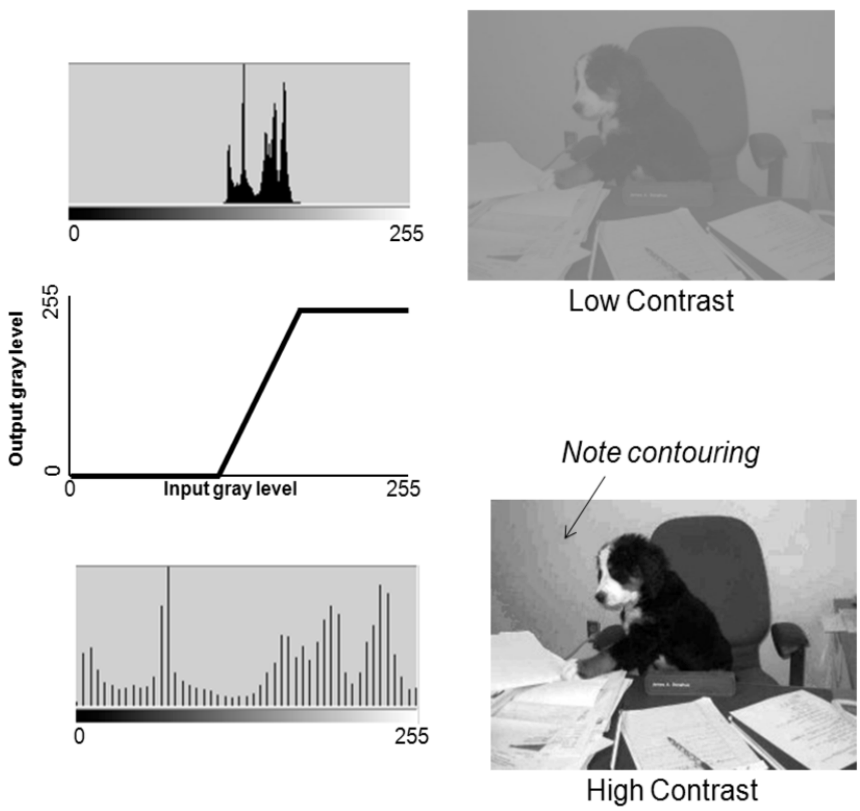

Figure 4: Enhancement by contrast stretch

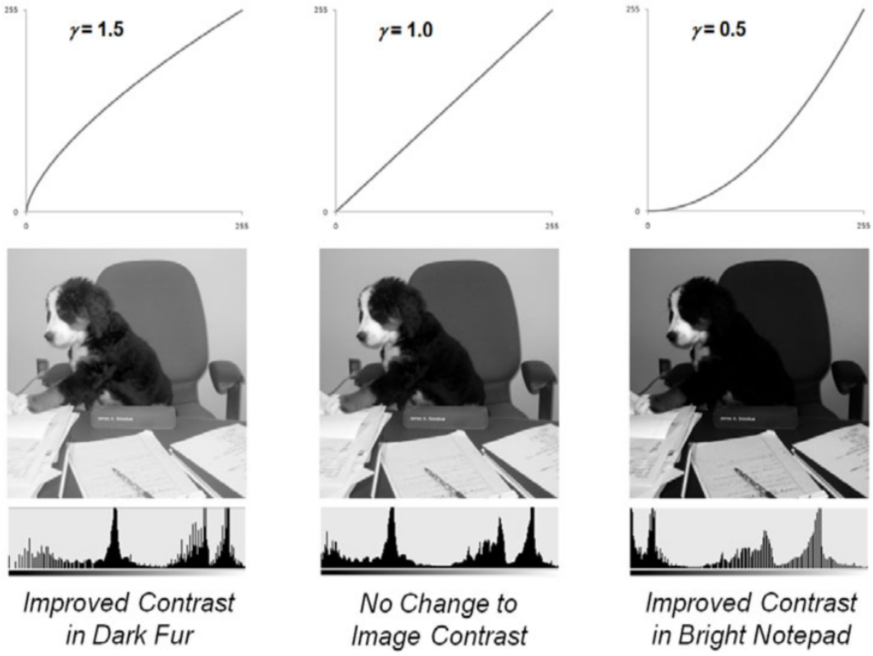

Figure 5: Enhancement using gamma correction

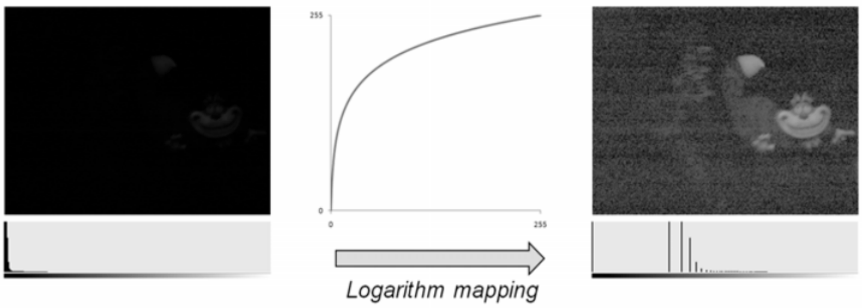

Figure 6: Logarithmic enhancement

If we want to enhance very dark images, the logarithmic enhancement method can be applied as

$$g'(x, y) = \lfloor 255 \cdot \frac{\log(g(x, y) + 1)}{255} \rfloor$$

Fig. 5 and Fig. 6 would show the effect of gamma correction and logarithmic enhancement.

Algorithm 1: Contrast Enhancement

Input : an Image I with $m \times n$ pixels
Output: the corresponding image after enhancement I'

hist $\leftarrow$ an array of 0s of size 256;
for $i = 1$ *to* m **do**
    for $j = 1$ *to* n **do**
        $GL \leftarrow$ the grey-level value $g(i, j)$;
        hist$[GL]$ $++$;
    end for
end for
Choose a specific method of enhancement M;
hist_after $\leftarrow$ an array of 0s of size 256;
for $i = 1$ *to* m **do**
    for $j = 1$ *to* n **do**
        $GL' \leftarrow M(g(i, j))$;
        $I'(i, j) \leftarrow GL'$;
        hist_after$[GL']$ $++$;
    end for
end for
return I';

References.

[1] *Contrast enhancement*. URL: http://printwiki.org/Contrast_Enhancement (cit. on p. 1).

[2] Robert D. Fiete. "Image Enhancement Processing". In: *Modeling the imaging chain of digital cameras*. SPIE Press, 2010, pp. 127–161 (cit. on pp. 1, 2).

[3] Charles A. Poynton. Morgan Kaufmann Publishers, 2003, p. 260 (cit. on p. 2).