## Image contrast and grey levels

The human eye can discern approximately 100 different brightness levels in any given image, although the actual range of sensitivity to brightness and dark is quite large. Thus any image display system should ~~st~~ represent images by a series of numbers from 0-100, at least. A common value used by many computers is 256 levels, or $2^8$. This is particularly convenient because one common computer storage unit is the byte, which itself can store values from 0 to 255. Hence in many image processing systems one byte is allocated for each pixel in the image.

For color images, one byte is often used for each of the red, green, and blue channels, for a total of 24 bits. Hence you will hear the term 24-bit color often used for color image formats. Since one word in many computers is 32 bits long, we often store a color image in part of a 32-bit word, with the remaining 8 bits available for other information.

## 8-bit displays

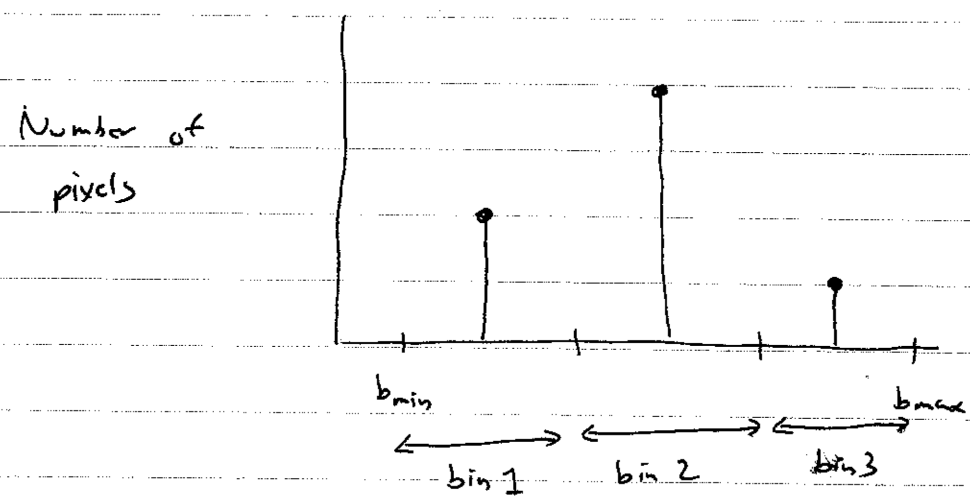
A computer display is often designed so that if the digital value assigned to a pixel is zero, the intensity is zero and the observed intensity is black. 255 corresponds to full intensity, or white. Each intermediate level is a value of gray ranging from very dark to very bright. This is somewhat more than the number of grey levels we can distinguish easily.

To display an image using the full range of the computer, then, we want the image to be displayed by numbers ranging from 0 to 255. Less of a range results in an image that is too dark, too bright, or otherwise restricted in brightness.

But an image may be generated by a device that outputs, say, values ranging from 0 to 1 for black to white. Such an image needs to be scaled to properly fit the display.

## Histograms

We use a histogram to describe the range of brightnesses in an image. The histogram simply gives us the number of pixels in a certain range of brightness. Pretend that we have an image function $f(x,y)$ that ranges from some minimum brightness $b_{min}$ to a maximum $b_{max}$. Then break up the interval into some number of "bins", or ranges of brightness. Plot in graphical form the number of pixels in each bin, as



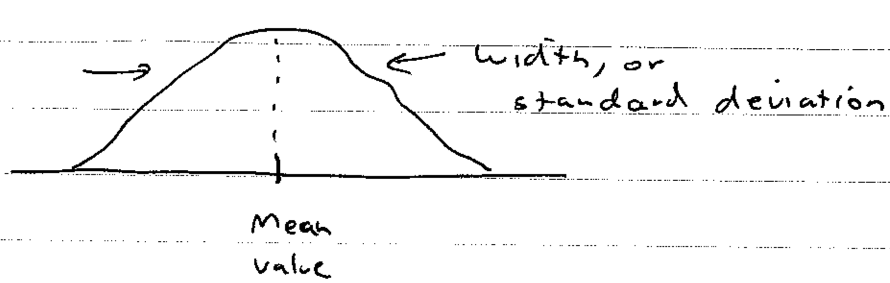

A sample histogram.

In this image most pixels are in bin 2, followed by half that number in bin 1, and $\frac{1}{4}$ that number in bin 3.

Because we will be scaling our image to a 256-level display, ~~to~~ we often will choose 256 bins in our histogram, although this choice is arbitrary. If we are calculating the histogram of a digital image that already range from 0 to 255, it makes sense often to assign to each bin the number of pixels with exactly that value.

What do the values expressed by the histogram tell us? While they do indicate the maximum and minimum values in the image, far more importantly they tell use the average value of brightness and how "spread" about that value the image is. The spread tells us how well the image uses the full range of the display.

We quantify the above two quantities by calculating the <u>mean</u>, or <u>average</u>, of the histogram, and also its <u>standard deviation</u>. These are related to what are called the <u>moments</u> of the function, or distribution.




We calculate the mean by determining the average value of the pixels. Let $f_{i,j}$ be the $(i,j)$th pixel in ~~an~~ image, where $i$ goes from 1 to $i_{max}$, and $j$ from 1 to $j_{max}$. Then the mean is the sum

$$\text{mean } (\mu) = \frac{\sum_{i=1}^{i_{max}} \sum_{j=1}^{j_{max}} f_{i,j}}{i_{max} \cdot j_{max}}$$

Note we use the symbol $\mu$ for the mean. We simply add all the values and divide by the number of points to obtain the mean.

Standard deviation is a bit more complicated, and there are a couple of formulas available to use. For large values of $i_{max} \cdot j_{max}$, the standard deviation (noted $\sigma$) is about

$$\sigma \approx \left[ \frac{\sum_{i=1}^{i_{max}} \sum_{j=1}^{j_{max}} (f_{i,j} - \mu)^2}{i_{max} \cdot j_{max}} \right]^{1/2}$$

This follow from definition of the variance, or square of the standard deviation, as the average of the square of the difference of each pixel from the mean value.

Note that if each $f_{i,j}$ is very close to the mean, then the standard deviation $\sigma$ is near zero. If the pixels differ significantly from the mean, the $\sigma$ is large.

Calculating $\sigma$ with the above formula is clumsy, however, as first we must calculate $\mu$, then subtract it from every point in the image, square each, and take the average. An easier method is ~~the~~ to use the following relation

$$\sigma = \left(M_2 - \mu^2\right)^{1/2}$$

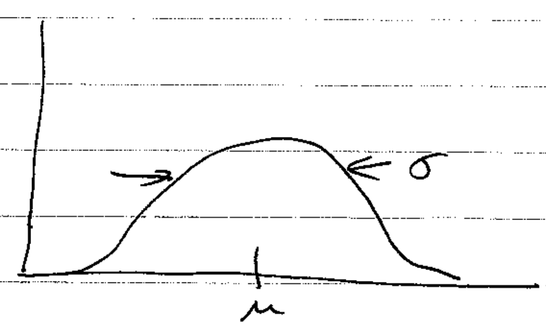
where $M_2$ is the second moment of the distribution, that is the average of the square of each pixel, or

$$M_2 = \frac{\sum_{i=1}^{i_{max}} \sum_{j=1}^{j_{max}} f_{i,j}^2}{i_{max} \cdot j_{max}}$$

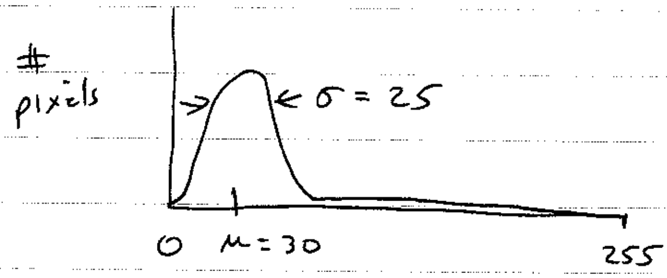
In this approach we calculate the average squared value of the function, subtract the square of the mean, and take the square root of the result.

Back to images - stretches

So, we can describe an image by its mean and standard deviation in the following picture:



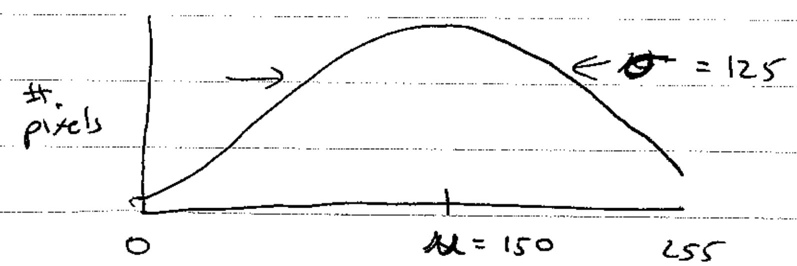

Suppose we have an image with a histogram that looks like the following:




In this image, the average value is 30 and the standard deviation 25. There are a few very bright pixels with values up to 255. How will this image look when displayed on the computer screen?

Recall that 0 is black and 255 white. Almost all of the points will be quite dark, with only a few bright points showing up. While the image will be an accurate depiction of the data, it will be hard to see and not very useful to us.

Suppose before displaying the image, we multiply each value by 5. This is called <u>stretching</u> the image because we have widened its distribution, which now looks like




The average value of the image is now 150, with $\sigma = 125$. This is a much brighter image than before, and it more evenly fills all the grey levels.

It is clear that $\mu$ and sigma scale linearly with multiplication by a constant:

$$\mu = \frac{\sum\sum f_{i,j}}{i_{max} \cdot j_{max}}$$

Scaled by a:

$$\mu' = \frac{\sum\sum a \cdot f_{i,j}}{i_{max} \cdot j_{max}} = a \frac{\sum\sum f_{i,j}}{i_{max} \cdot j_{max}} = a \cdot \mu$$

For $\sigma$:

$$\sigma = \left[ \frac{\sum\sum f_{i,j}^2}{i_{max} \cdot j_{max}} - \left( \frac{\sum\sum f_{i,j}}{i_{max} \cdot j_{max}} \right)^2 \right]^{1/2}$$

$$\sigma' = \left[ \frac{\sum\sum a^2 f_{i,j}^2}{i_{max} \cdot j_{max}} - \left( \frac{\sum\sum a f_{i,j}}{i_{max} \cdot j_{max}} \right)^2 \right]^{1/2}$$

$$= \left[ \frac{a^2 \sum\sum f_{i,j}^2}{i_{max} \cdot j_{max}} - a^2 \left( \frac{\sum\sum f_{i,j}}{i_{max} \cdot j_{max}} \right)^2 \right]^{1/2}$$

$$= a \cdot \sigma$$

We don't have to restrict ourselves to a simple scaling. Note that we can shift both the mean and standard deviation independently, as multiplication changes both $\mu$ and $\sigma$, and we can always add a constant value to shift $\mu$ without changing sigma.

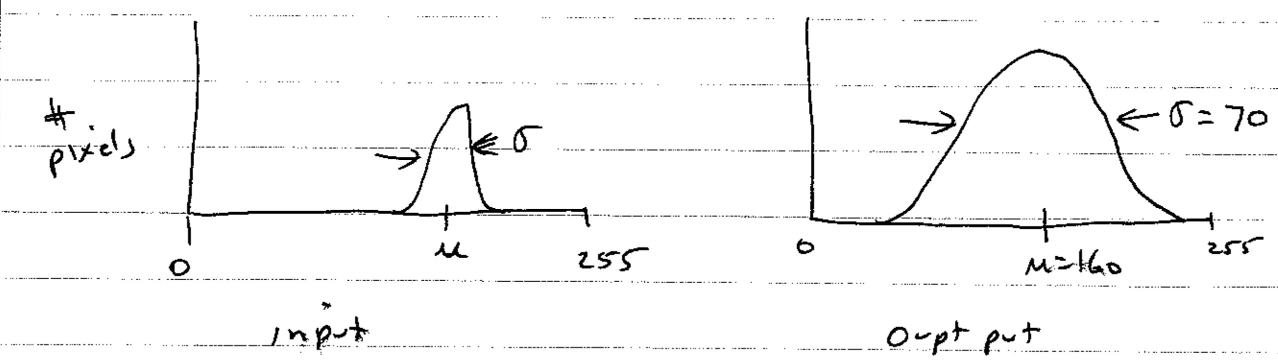
Also, if $\mu = 0$, then any scaling only affects $\sigma$.

The usual recipe is

1) Shift the input image to zero mean by subtracting $\mu$ from every point
2) Multiply by a constant to set the new $\sigma$
3) Add in any constant we like for the new $\mu$

A good rule of thumb is to give images a mean value of about 160 and a standard deviation of 60-80 for an eight-bit display.

So, typical input and output histograms before and after stretching might look like



A little reflection lets us define a transformation on the image to stretch a given image to a desired new mean and standard deviation:

$$g(x,y) = \frac{\sigma'}{\sigma}\left(f(x,y) - \mu\right) + \mu'$$

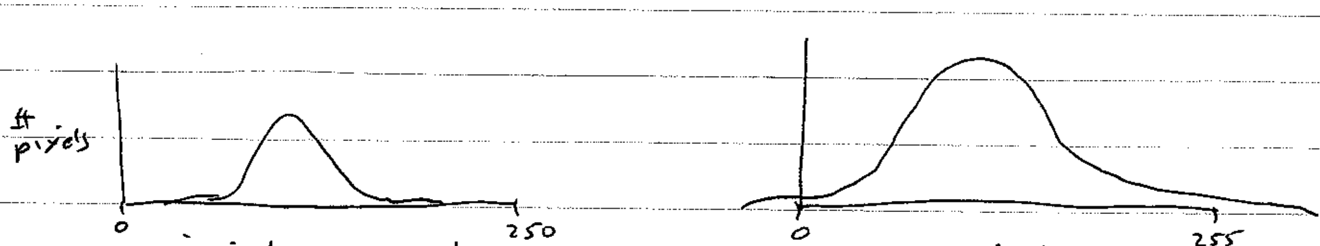
where $\sigma, \mu$ represent the original image and $\sigma', \mu'$ the new image. Multiplying out gives

$$g(x,y) = \frac{\sigma'}{\sigma} f(x,y) - \frac{\sigma'}{\sigma}\mu + \mu'$$

which is of the form $g = a \cdot f + b$, or a linear function. Hence we call this kind of stretch a linear stretch.

## Ends of the distribution

If we stretch an image, in many cases pixels located far from the mean end up with new, calculated values that are either greater than 255 or less than 0. What do we do about these?




original image histogram

stretched histogram

Since the display can only assign grey levels that lie between 0 and 255, we must have a rule to deal with the out liers. In most cases, to form a pleasing image, pixels that are calculated to be blacker than black (zero) or whiter than white (255) can simply be set to the limits of the display. Our rule for stretching an image thus becomes

$$g(x,y) = a \cdot f(x,y) + b \qquad \text{for } 0 \leq g(x,y) \leq 255$$

$$= 0 \qquad \text{for } g < 0$$

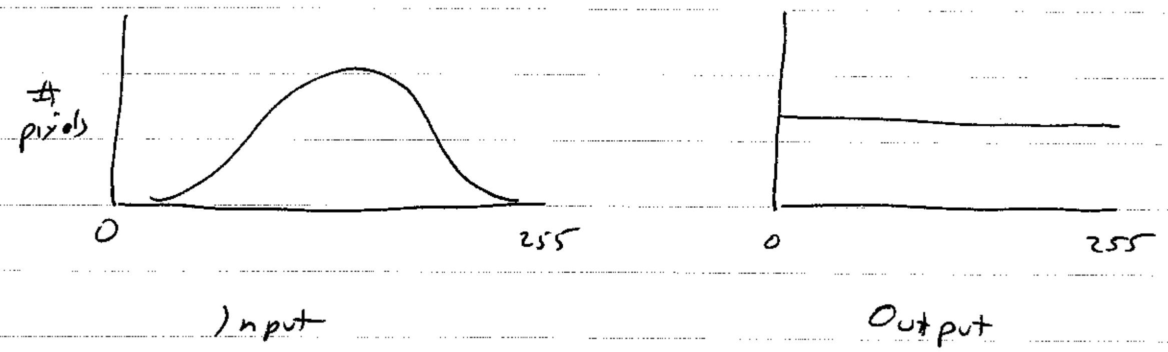
$$= 255 \qquad \text{for } g > 255$$

Sometimes we find we have to limit the amount we stretch a particular image so that too many pixels are not stretched off the ends of the distribution.

<u>Arbitrary stretches</u>

Linear stretches are very useful for matching an image to a display. However, for certain visual effects, or even to make an image appear more pleasing, we might find that we want to force the output histogram to have a specific shape. We can derive a transformation to map any slice distribution to what we want.

The approach is perhaps most easily seen if we look at an example where the output histogram is approximately flat -- we call this instance <u>histogram equalization</u>. The term comes from the property that the number of output pixels in each output bin level is about equal. The input and output histograms might look like this:

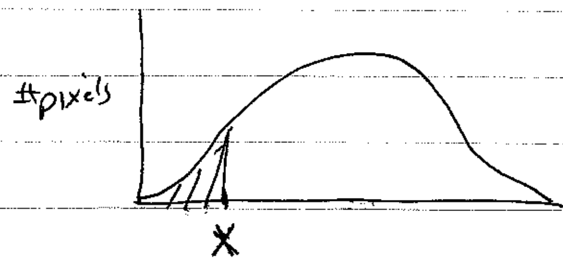



To derive the transformation, note that we want the $\frac{N}{256}$ darkest pixels to map into output level 0, where $N$ is the total number of pixels in the image. Then the $\frac{N}{256}$ next brightest map into level 1, and so on until the brightest $\frac{N}{256}$ pixels map to 255.

That takes care of the output. How do we find the $\frac{N}{256}$ darkest pixels? Note that the total number of pixels $N$ is the integral of each histogram. Hence if we numerically integrate the input curve until we get $N/256$, we can identify the range of inputs that map into level 0, by the integral relation

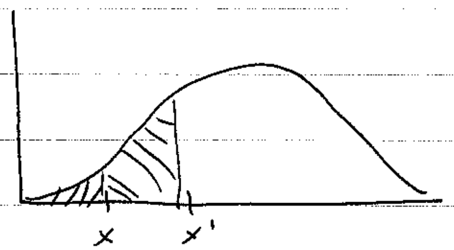
$$\int_0^X h(x)\,dx = \frac{N}{256}$$

where $h(x)$ is the histogram and $X$ is the top of the interval:



The shaded area is equal to $N/256$. Then we find the next interval such that

$$\int_0^{x'} h(x)\,dx = 2 \cdot \frac{N}{256}$$




and repeat until all intervals are determined. Then the mapping is defined as

$$\begin{aligned} g(x,y) &= 0 \qquad \text{if } 0 \le f(x,y) \le x \\ &= 1 \qquad \text{if } x < f(x,y) \le x' \\ &\ \ \vdots \qquad\qquad\quad \vdots \end{aligned}$$

and so forth.

Complications here in the actual implementation due to a finite number of pixels and quantized input values. The integrals must be approximated by sums of the input bins, which usually won't equal the exact limits we want. Hence we add successive bins starting from zero until we meet or exceed the desired thresholds, and assign outputs accordingly. The result of this is that some output bins are never used, and the "equalized" histogram is never exactly flat. But it is close enough for most imaging applications.

How about if we want an arbitrary shape rather than a flat, equalized histogram? For example, one common stretch is called a Gaussian stretch, where the output histogram is constrained to have the form of a Gaussian function

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

where $\mu$ and $\sigma$ as usual denote the mean and standard deviation of the distribution. For the Gaussian stretch, we don't integrate until the bins are $\frac{i}{256}$, $0 \leq i \leq 256$, but until the limit is the integral of the output distribution over appropriate limits, as in

$$\text{For bin 0:} \quad \int_0^x h(x)\,dx = \int_{-\infty}^{\text{bin 1 limit}} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx$$

Note that for bin 0, we integrate from $-\infty$ rather than 0 as the Gaussian distribution extends to $\infty$ in both directions. Then for bin 1,

$$\int_0^{x'} h(x)\,dx = \int_{-\infty}^{\text{bin 2 limit}} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\pi\sigma^2}} dx$$

and so forth.

Once again, we have quantization effects that allow only approximate solutions. Nonetheless many people think a pseudo-Gaussian image is very pleasing, so we see this stretch quite often.