Histogram Equalization

Carlo Tomasi

Let $I(\mathbf{x})$ be a gray-level image with n pixels and with values in $\mathcal{V} = \{0, \ldots, v_{\max}\}$ and let $\mathcal{P}(\mathbf{x})$ be a pixel predicate. The number of pixels that satisfy the predicate is denoted by $N(\mathcal{P})$. The *histogram* of I is the function $h_I : \mathcal{V} \to \mathbb{N}$ defined by

$$h_I(u) = N(I(\mathbf{x}) = u)$$

and the *cumulative count* of I is the function $H_I : \mathcal{V} \to \mathbb{N}$ defined by

$$H_I(u) = N(I(\mathbf{x}) \leq u) = \sum_{i \leq u} h_I(i)$$

so that

$$h_I(u) = \begin{cases} H_I(u) & \text{for } u = 0 \\ H_I(u) - H_I(u-1) & \text{otherwise} \end{cases} .$$

Let the function

$$f : \mathcal{V} \to \mathcal{V}$$

be some point transformation of the image:

$$J(\mathbf{x}) = f(I(\mathbf{x})) .$$

Then, the cumulative count of the transformed image J is

$$H_J(v) = N(J(\mathbf{x}) \leq v) = N(f(I(\mathbf{x})) \leq v) .$$

If f is strictly monotonic and increasing, then it is invertible and

$$H_J(v) = N(I(\mathbf{x}) \leq f^{-1}(v)) = H_I(f^{-1}(v)) . \tag{1}$$

Equalizing the histogram of $I(\mathbf{x})$ amounts to applying a point transformation f to it so that

$$h_J(v) \approx c \quad \text{so that} \quad H_J(v) \approx [v+1]c . \tag{2}$$

where

$$c = \frac{n}{|\mathcal{V}|} \ .$$

Equations (1) and (2) show that histogram equalization requires f to satisfy

$$H_I(f^{-1}(v)) \approx [v+1]c$$

so that

$$\frac{1}{c} H_I(f^{-1}(v)) - 1 \approx v \ .$$

This result shows that f^{-1} is the approximate inverse of the function

$$g(u) = \frac{1}{c} H_I(u) - 1 \ ,$$

so

$$f(u) \approx g(u) = \frac{1}{c} H_I(u) - 1 \ . \tag{3}$$

This derivation assumes that f, and therefore the cumulative count H_I of the input image, is strictly monotonic. If it is not, the definition (3) can still be used, but the histogram of the resulting image will be farther away from constant.

A simple equalization function (that also optionally returns f) can thus be written as follows in MATLAB:

```
function[J, f] = equalize(I)
vmax = double(intmax(class(I)));
h = hist(I(:), 0:vmax);
H = cumsum(h);
c = numel(I) / (vmax + 1);
f = H/c - 1;
J = cast(f(I), class(I));
```

Regardless of the nature of H_I, exact equalization can generally not be achieved with a point transformation. The fundamental reason for this is that a point transformation $v = f(u)$ maps *every* pixel whose value is u to the new value v. In terms of histograms, this means that the histogram bar $h_I(u)$ can only be moved *in toto* to a different position by the change of variable $h_J(v) = h_I(f^-1(v))$. Different bars $h_I(u_1), \ldots, h_I(u_k)$ can be moved to the same position v, in which case

$$h_J(v) = \sum_{i=1}^{k} h_I(u_k) \ .$$

The additional requirement that f be monotonic and increasing implies that remapped values preserve order,

$$u_1 < u_2 \Rightarrow v_1 \leq v_2$$

so the ordering of two bars in h_I cannot be reversed in h_J.

Because of this, *the bars in the new histogram are the same bars as in the old histogram, spread out in a different way, and with the possibility of collision (two or more bars moving to the same bin of h_J and adding up their values as a result)*. This is a very strong constraint on what histograms can be obtained by a point transformation. The example in Figure 1 may help clarify. While the detailed histogram of the output image is not constant, a histogram with much wide bins is roughly constant. The gaps in the detailed histogram of the output image (visible when the plot is displayed with enough magnification) are values where $H_J(v-1) = H_J(v)$, so that $h_J(v) = H_J(v) - H_J(v-1) = 0$.

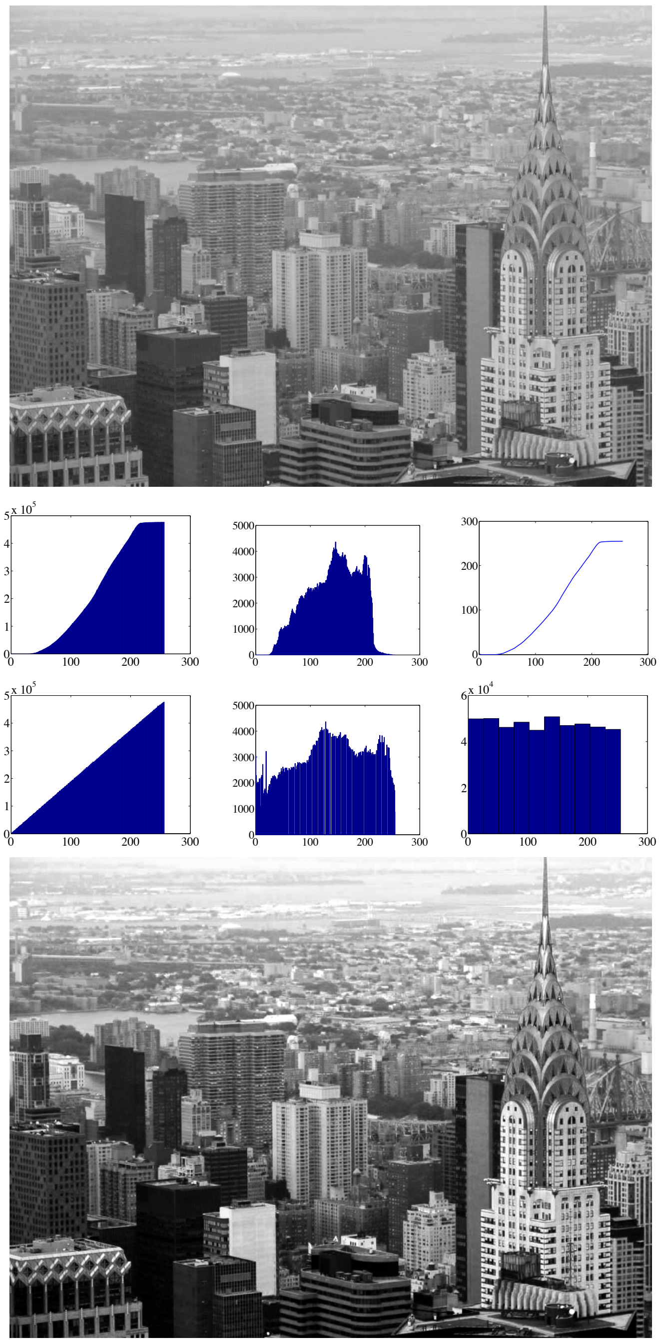

Figure 1: The three plots under the input image at the top are its cumulative count, histogram, and the equalization function f. The three plots above the equalized image at the bottom are its cumulative count, histogram, and a histogram with coarse bins.