Image Pre-processing

Image Pre-processing

- Histogram equilization
- Geometric transformations
- Corners

Histogram Equilization

$$H(p) = \#\{I(i,j) = p\}, \forall i, j, p$$

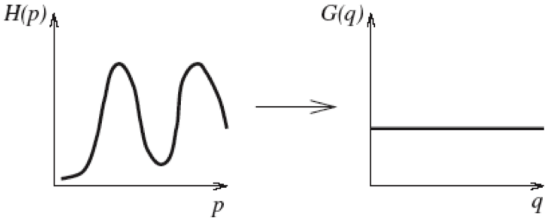


Figure 5.2: Histogram equalization.

$$\sum_{i=0}^{k} H(p_i) = \sum_{i=0}^{k} G(q_i) = N \times M$$

Histogram Equilization (cont.)

For each histogram bin we should have

$$f = \frac{N \times M}{q_k - q_0}$$

We are looking for a transformation $q = \mathcal{T}(p)$ such that

$$NM \int_{q_0}^{q} \frac{1}{q_k - q_0} ds = \frac{NM(q - q_0)}{q_k - q_0} = \int_{p_0}^{p} H(s) ds$$

$$q = \mathcal{T}(p) = \frac{q_k - q_0}{NM} \int_{p_0}^{p} H(s) ds + q_0$$

The integral is called a cumulative histogram. A discrete approximation is

$$q = \mathcal{T}(p) = \frac{q_k - q_0}{NM} \sum_{i=p_0}^{p} H(i) + q_0$$

Algorithm for Histogram Equilization

Algorithm 5.1: Histogram equalization

1. For an $N \times M$ image of G gray-levels (often 256), create an array H of length G initialized with 0 values.
2. Form the image histogram: Scan every pixel and increment the relevant member of H—if pixel p has intensity g_p, perform

$$H[g_p] = H[g_p] + 1\,.$$

3. Form the cumulative image histogram H_c:

$$H_c[0] = H[0]\,,$$
$$H_c[p] = H_c[p-1] + H[p]\,, \quad p = 1, 2, \ldots, G-1\,.$$

4. Set

$$T[p] = \text{round}\left(\frac{G-1}{NM} H_c[p]\right)\,.$$

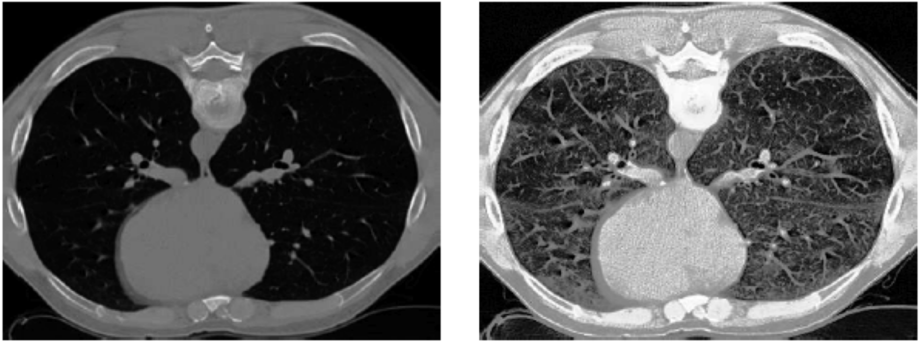
(This step obviously lends itself to more efficient implementation by constructing a look-up table of the multiples of $(G-1)/NM$, and making comparisons with the values in H_c, which are monotonically increasing.)

5. Rescan the image and write an output image with gray-levels g_q, setting

$$g_q = T[g_p]\,.$$

Histogram Equilization: Example

Improved dynamic range:



(a) (b)

Figure 5.3: Histogram equalization. (a) Original image. (b) Equalized image.

Histogram Equilization: Histograms

Improved dynamic range:

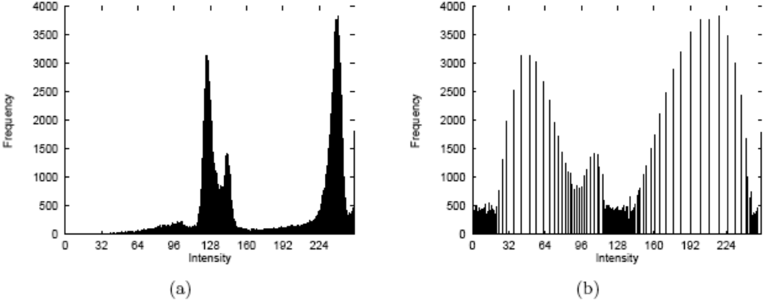


Figure 5.4: Histogram equalization: Original and equalized histograms corresponding to Figure 5.3a,b.

Geometric Transformations

$$x' = T_x(x, y), \quad y' = T_y(x, y)$$

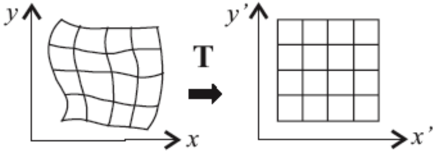


Figure 5.5: Geometric transform on a plane.

Approximate the above transform using a *bilinear transformation*

$$\begin{aligned} x' &= a_0 + a_1 x + a_2 y + a_3 xy \\ y' &= b_0 + b_1 x + b_2 y + b_3 xy \end{aligned}$$

Can be done using four pairs of corresponding points

Geometric Transformations

Affine transformation is simpler and needs only three correspondences

$$\begin{aligned} x' &= a_0 + a_1 x + a_2 y \\ y' &= b_0 + b_1 x + b_2 y \end{aligned}$$

Typical geometric transformations such as *rotation*, *translation*, *scaling*, and *skewing* are included

Jacobian matrix J provides the information how the coordinates change

$$J = \frac{\delta(x', y')}{\delta(x, y)} = \begin{pmatrix} \frac{\delta x'}{\delta x} & \frac{\delta x'}{\delta y} \\ \frac{\delta y'}{\delta x} & \frac{\delta y'}{\delta y} \end{pmatrix}$$

Determinant of J shows the change of scale because of the transform

For bilinear transform

$$\det(J) = a_1 b_2 - a_2 b_1 + (a_1 b_3 - a_3 b_1)x + (a_3 b_2 - a_2 b_3)y$$

For affine transform

$$\det(J) = a_1 b_2 - a_2 b_1$$

Important Geometric Transformations

Rotation by the angle ϕ about the origin

$$\begin{aligned} x' &= x\cos\phi + y\sin\phi \\ y' &= -x\sin\phi + y\cos\phi \\ \det(J) &= 1 \end{aligned}$$

Change of scale a in the x axis and b in y axis

$$\begin{aligned} x' &= ax \\ y' &= by \\ \det(J) &= ab \end{aligned}$$

Skewing by the angle ϕ

$$\begin{aligned} x' &= x + y\tan\phi \\ y' &= y \\ \det(J) &= 1 \end{aligned}$$

Distortion Types

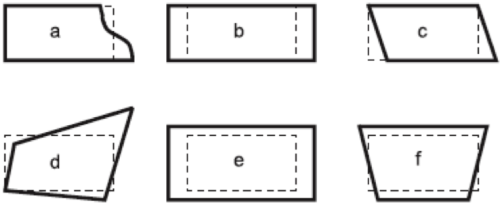


Figure 5.6: Geometric distortion types.

Non-linearity (a), panoramic distortion (b), skew (c), scale (e), perspective (f)

Brightness Interpolation

Invert the transform

$$(x, y) = T^{-1}(x', y')$$

Nearest neighbor interpolation:

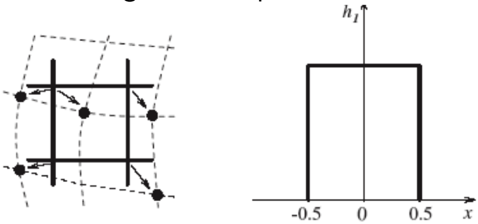


Figure 5.7: Nearest-neighborhood interpolation. The discrete raster of the original image is depicted by the solid line.

$$f_1(x, y) = g_s(round(x), round(y))$$

Brightness Interpolation

Bilinear interpolation:

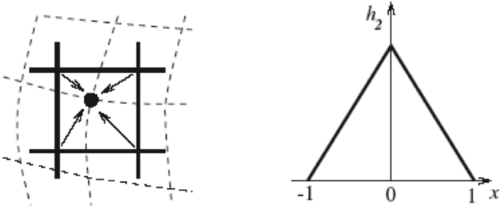


Figure 5.8: Linear interpolation. The discrete raster of the original image is depicted by the solid line.

$$
\begin{aligned}
f_2(x,y) &= (1-a)(1-b)g_s(l,k) + a(1-b)g_s(l+1,k) \\
&\quad + b(1-a)g_s(l,k+1) + ab g_s(l+1,k+1)
\end{aligned}
$$

where

$$l = floor(x), \quad a = x - l, \quad k = floor(y), \quad b = y - k$$

Brightness Interpolation

Bicubic interpolation:

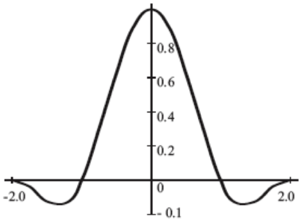


Figure 5.9: Bi-cubic interpolation kernel.