GRAPHICS GROUP

Intelligent Vision Systems New Zealand

Computer Graphics and Image Processing

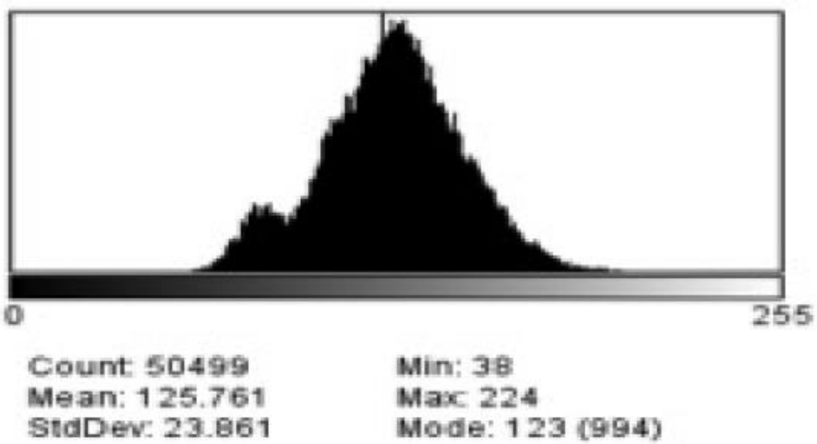
Part 3: Image Processing

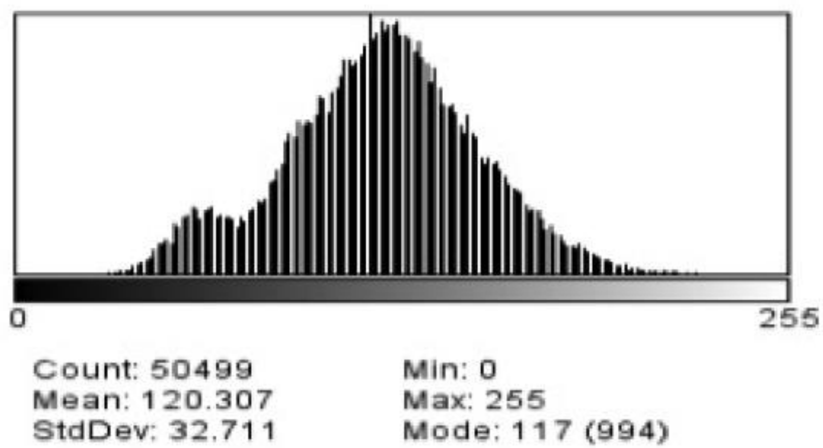
3 – Histogram Equalisation

Martin Urschler, PhD

Contrast stretching example

$f_{\text{low}} = f_{\min} = 38;\ f_{\text{high}} = f_{\max} = 224;\ g_{\min} = 0;\ g_{\max} = 255:$

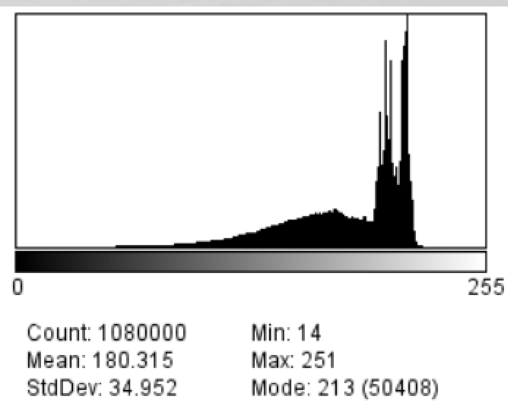




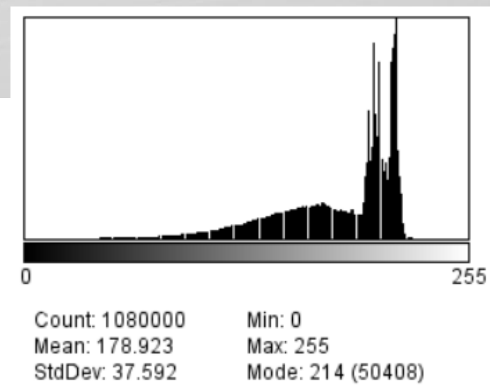
$$g(x,y) = (f(x,y) - 38)\,\frac{255-0}{224-38} = 1.37 f(x,y) - 52.1$$

Contrast stretching example

- Often histogram has values nearly between full range, making simple contrast stretching ineffective



Before
contrast stretching



After
contrast stretching

Percentile Based Mapping

Percentile (see also: `http://en.wikipedia.org/wiki/Percentile`)

The pixel-wise intensity below which a certain percent of pixels fall.

The $\alpha - \beta$ percentile range $\Rightarrow$ to the max range $0 - 255$:

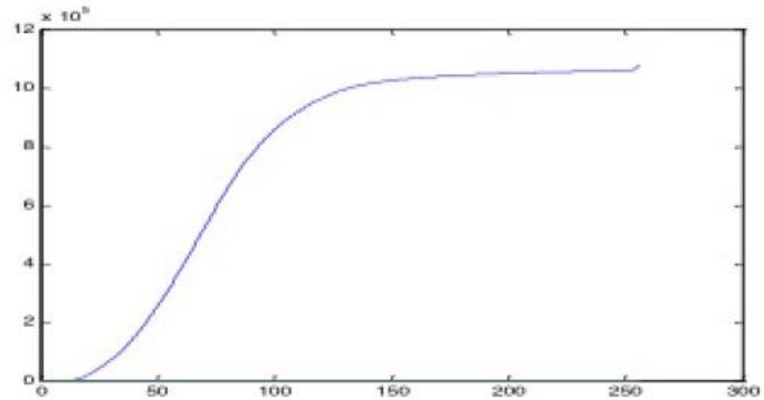
1. Collect the histogram $H = (H(q) : q \in \mathbf{Q})$ and compute the $\boxed{\text{cumulative histogram } C} = (C(q) : q \in \mathbf{Q})$; $C(q) = \sum_{j=0}^{q} H(j)$.
2. Find the smallest value, q_α, such that $C(q_\alpha)$ is larger than $\alpha\%$ of the overall number K of pixels.
3. Find the largest value, q_β, such that $C(q_\beta)$ is smaller than $\beta\%$ of the overall number K of pixels.
4. Perform linear mapping: $g(x, y) = \frac{255}{q_\beta - q_\alpha}(f(x, y) - q_\alpha)$.
   - Keep values $g(x, y)$ below 0 at 0 and above 255 at 255.

Cumulative histogram?

$$H = [40, 0, 18, 6, 10, 3, 7, 16] \;\rightarrow C = [40, 40, 58, 64, 74, 77, 84, 100]$$

A cumulative histogram $C = [C(q) : q = 0, \ldots, 255]$

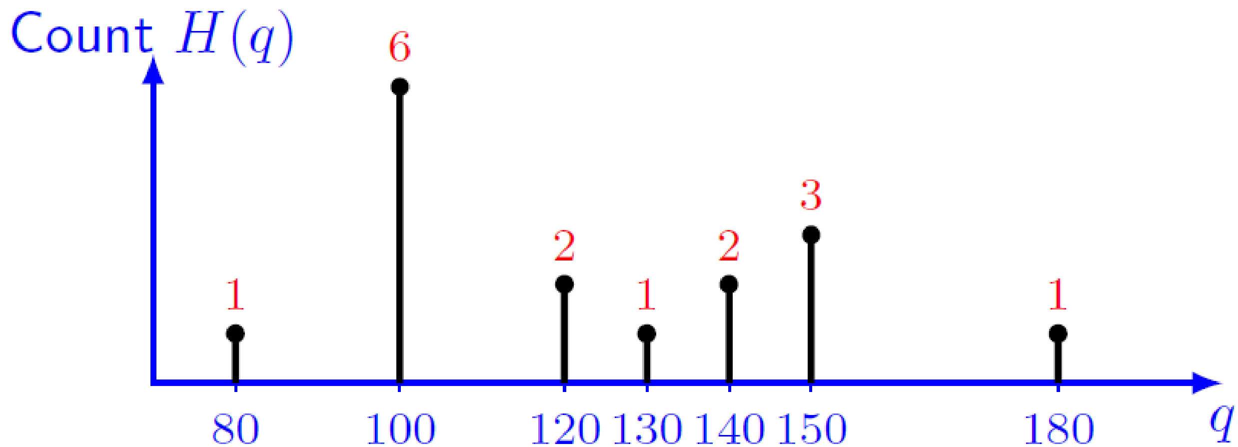
is a mapping that counts the total number of pixel intensities in all the histogram's bins up to the current bin q: $C(q) = \sum_{i=0}^{q} H(i)$.



The cumulative histogram is useful for some pixel-wise intensity corrections, e.g. percentile based mapping, histogram equalisation.

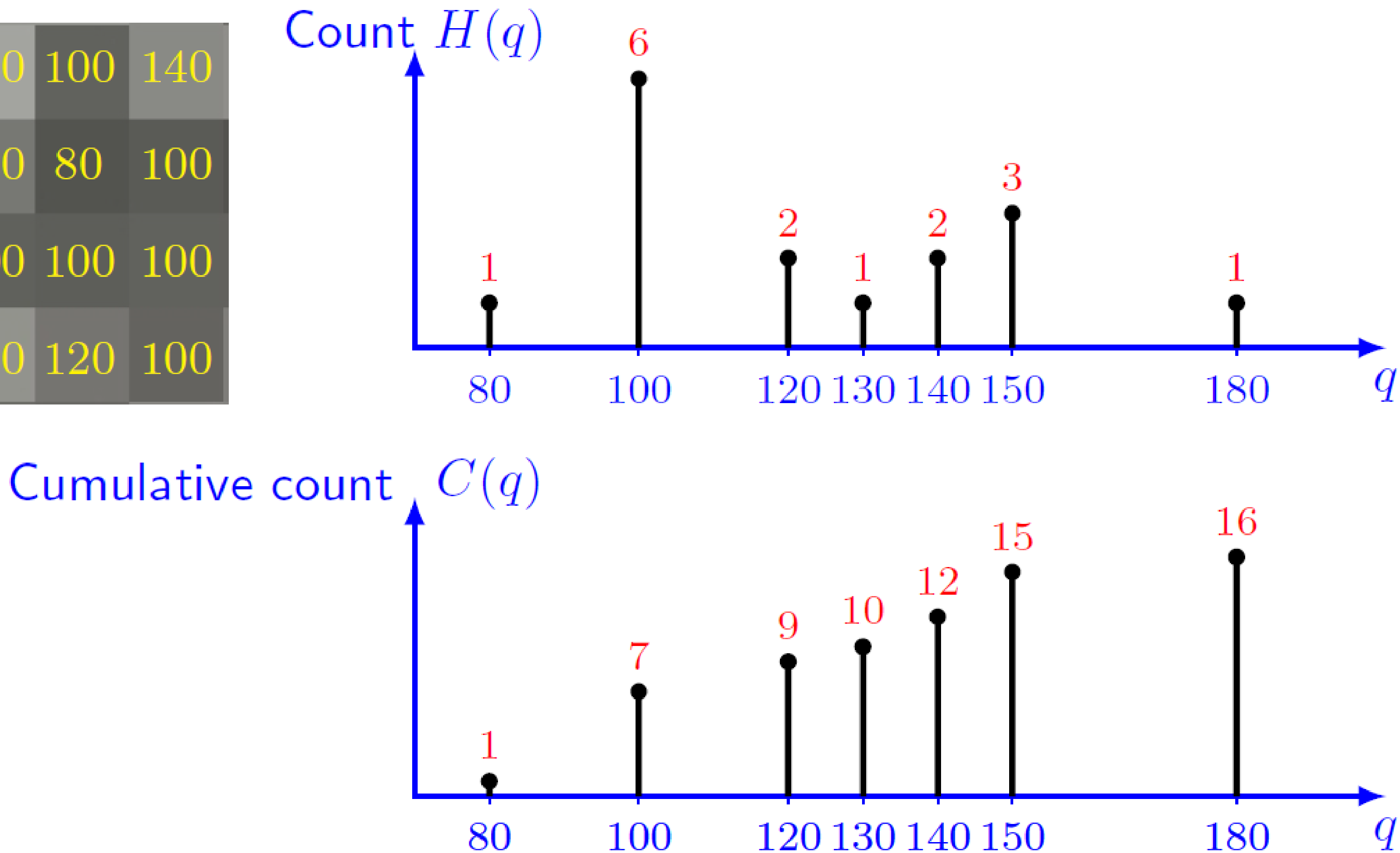
Cumulative histogram - example

180	150	100	140
140	120	80	100
130	100	100	100
150	150	120	100



Cumulative histogram - example

180	150	100	140
140	120	80	100
130	100	100	100
150	150	120	100



Percentile Based Mapping

Percentile (see also: `http://en.wikipedia.org/wiki/Percentile`)

The pixel-wise intensity below which a certain percent of pixels fall.

The $\alpha - \beta$ percentile range $\Rightarrow$ to the max range $0 - 255$:

1. Collect the histogram $H = (H(q) : q \in \mathbf{Q})$ and compute the cumulative histogram $C = (C(q) : q \in \mathbf{Q})$; $C(q) = \sum_{j=0}^{q} H(j)$.
2. Find the smallest value, q_α, such that $C(q_\alpha)$ is larger than $\alpha\%$ of the overall number K of pixels.
3. Find the largest value, q_β, such that $C(q_\beta)$ is smaller than $\beta\%$ of the overall number K of pixels.
4. Perform linear mapping: $g(x, y) = \frac{255}{q_\beta - q_\alpha}(f(x, y) - q_\alpha)$.
   - Keep values $g(x, y)$ below 0 at 0 and above 255 at 255.

Back to Percentile Based Mapping

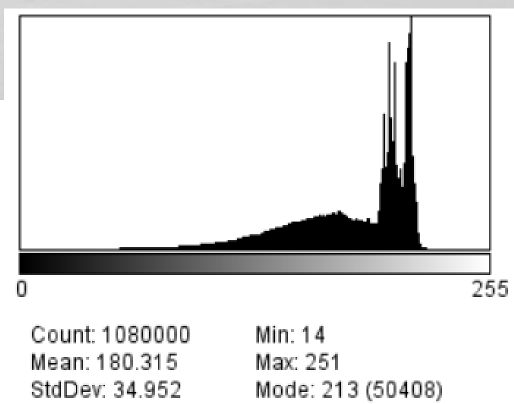
Percentile (see also: `http://en.wikipedia.org/wiki/Percentile`)

The pixel-wise intensity below which a certain percent of pixels fall.

The $\alpha - \beta$ percentile range $\Rightarrow$ to the max range $0 - 255$:

1. Collect the histogram $H = (H(q) : q \in \mathbf{Q})$ and compute the cumulative histogram $C = (C(q) : q \in \mathbf{Q})$; $C(q) = \sum_{j=0}^{q} H(j)$.
2. Find the smallest value, q_α, such that $C(q_\alpha)$ is larger than $\alpha\%$ of the overall number K of pixels.
3. Find the largest value, q_β, such that $C(q_\beta)$ is smaller than $\beta\%$ of the overall number K of pixels.
4. Perform linear mapping: $g(x, y) = \frac{255}{q_\beta - q_\alpha}(f(x, y) - q_\alpha)$.
   - Keep values $g(x, y)$ below 0 at 0 and above 255 at 255.

1 – 99 percentile mapping example

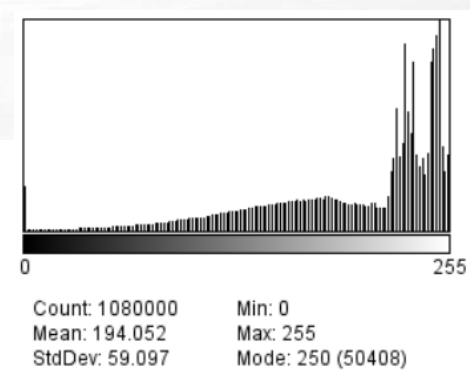


Original

After 1-99 percentile contrast stretching

$f_{low} = f_{1\%} = 67$; $f_{high} = f_{99\%} = 216$;

$g_{min} = 0$; $g_{max} = 255$



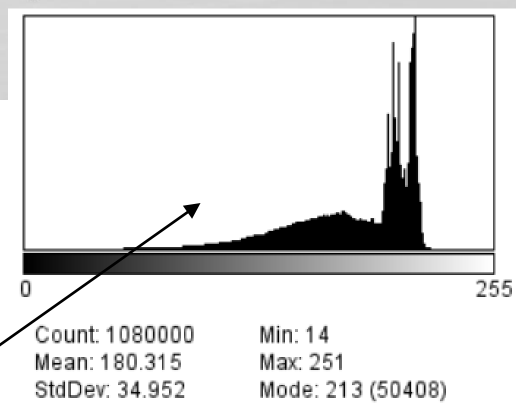
Histogram equalisation

- A non-linear mapping of pixel-wise intensities aimed at flattening the image histogram (distributing evenly the output intensities).
  - It **increases the dynamic range** and as a result increases image contrast.
  - It may be useful in images where foreground and background are both either bright or dark.
  - It tends to reveal details that would be otherwise hidden.
  - It often produces unrealistic effects in photographs, but is very useful in scientific (e.g. x-ray, satellite, or thermal) images.

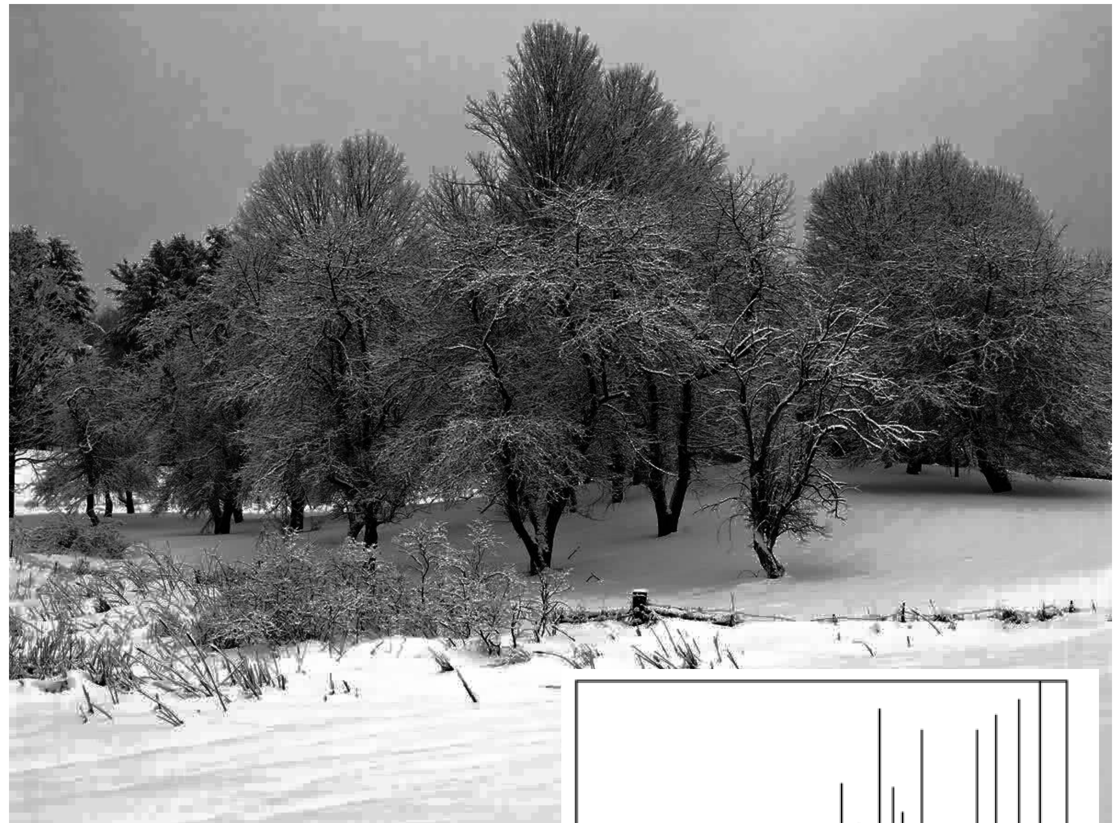
- It differs from contrast stretching in the use of non-linear transfer functions to map between the input and output intensities.
  - The mapping function is derived from the image histogram, and aims at **flattening the cumulative histogram** of the equalised image.

Histogram equalisation example

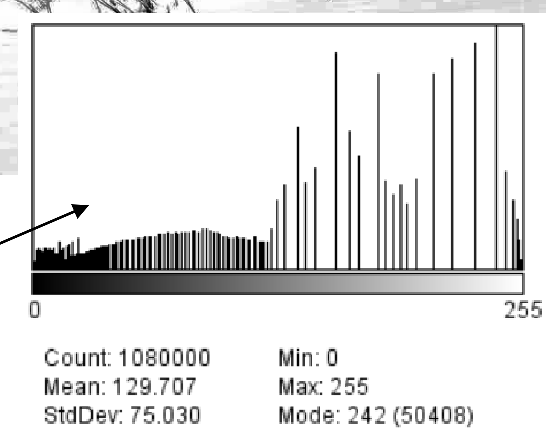
Original



Poor dynamic range



After histogram equalisation



More even distribution

Histogram equalisation algorithm

Given an image f and its histogram $H = (H(q) : q = 0, 1, \ldots, Q)$:

1: Compute the cumulative histogram C

for q = 1,...,Q do C[q] = C[q-1]+ H[q], $C[0] = H[0]$

2: Convert C into the LUT (lookup table) T

q_{min} : minimum q for which $C[q]$ is larger than 0

for q = 0,..., Q do
    if q < q_{min} $T[q] = 0$
    else T[q] = Q * (C[q] - C[q_{min}])$/(C[Q] - C[q_{min}])$
end for

Smallest q for which C[q] is equal to #pixels (K)

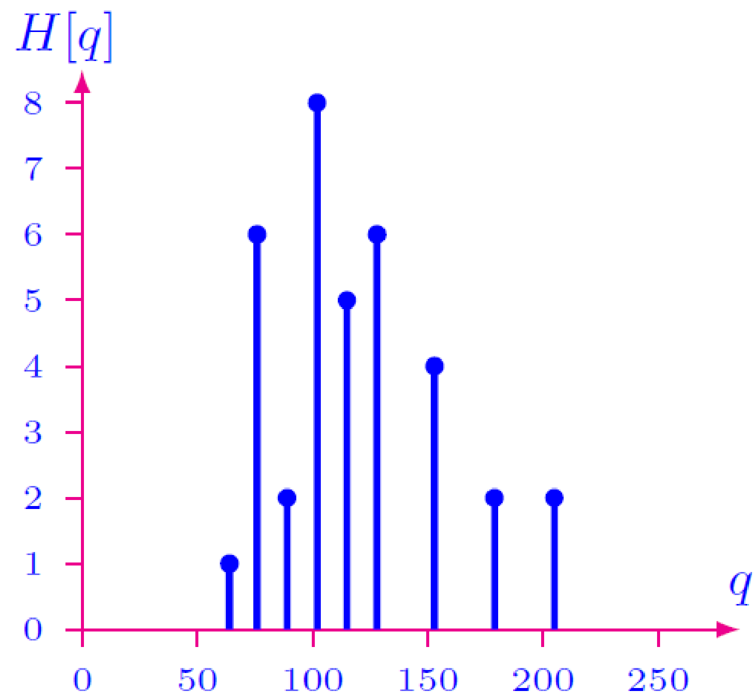
3. Using the LUT T, transform f into the equalised image g

for all pixels (x,y) do g[x,y] = T[f[x,y]]

Histogram equalisation: 6x6 example

Collecting the histogram H

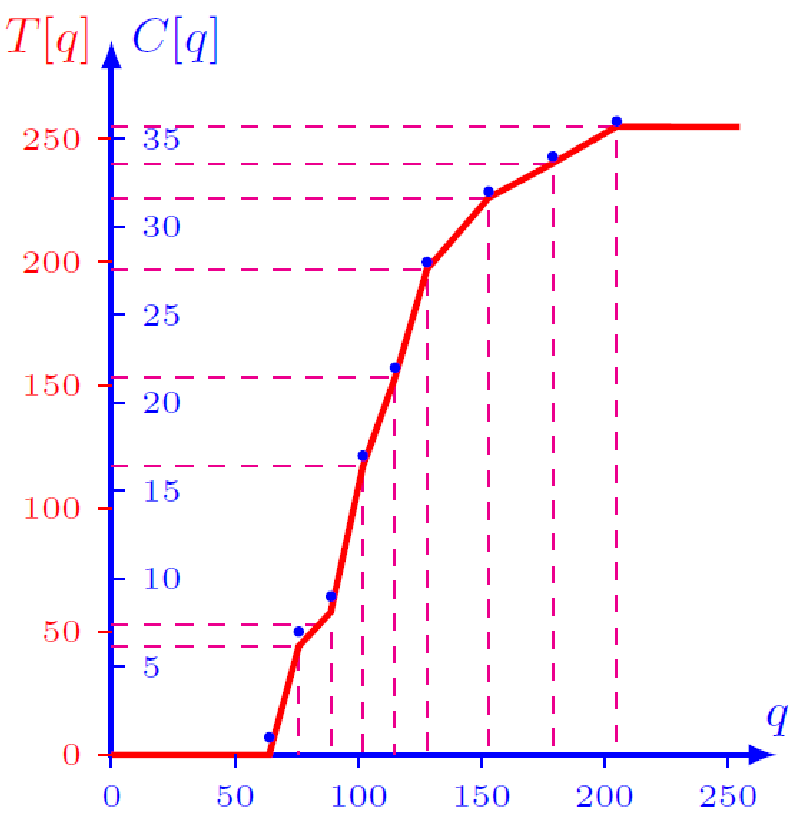
q	64	76	89	102	115	128	153	179	205
$H[q]$	1	6	2	8	5	6	4	2	2



Histogram equalisation: 6x6 example

Computing the cumulative histogram C and the LUT T

q	64	76	89	102	115	128	153	179	205
$H[q]$	1	6	2	8	5	6	4	2	2
$C[q]$	1	7	9	17	22	28	32	34	36
$T[q]$	0	44	58	117	153	197	226	240	255



$$
\begin{aligned}
C[q] &= \sum_{j=0}^{q} H[q] \\
T[q] &= \text{round}\left\{255 \cdot \frac{C[q]-C[64]}{C[205]-C[64]}\right\} \\
&= \text{round}\left\{255 \cdot \frac{C[q]-1}{36-1}\right\} \\
&= \text{round}\left\{7.286 \cdot (C[q]-1)\right\}
\end{aligned}
$$

---

round$\{z\}$ – the closest to z integer number: e.g. round$\{3.45\} = 3$ and round$\{3.51\} = 4$.

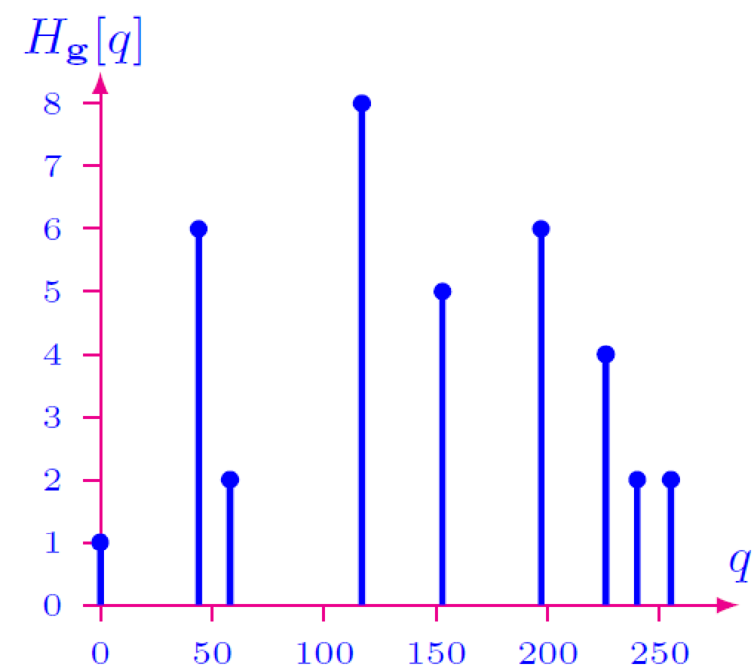
Histogram equalisation: 6x6 example

Transforming $\mathbf{f}$ in line with the LUT T: $g(x,y) = T[f(x,y)]$.

0	117	117	117	44	44
117	226	255	197	153	197
117	226	255	240	226	197
153	197	226	240	117	153
44	117	197	153	117	197
58	117	44	58	44	44

q	0	44	58	117	153
$H_{\mathbf{g}}[q]$	1	6	2	8	5

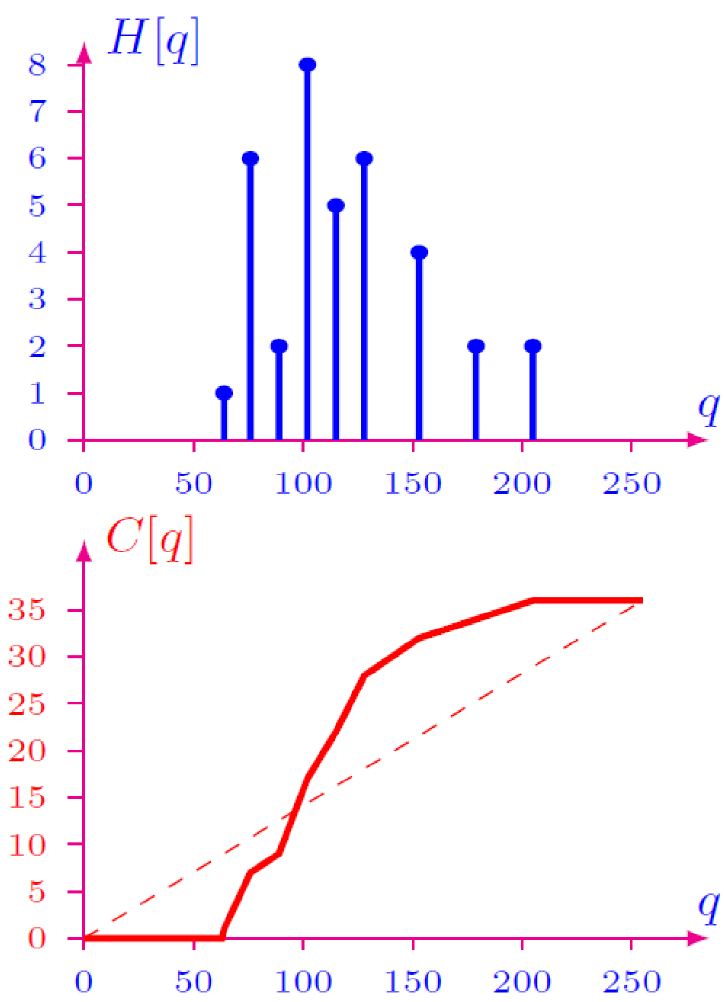
q	197	226	240	255
$H_{\mathbf{g}}[q]$	6	4	2	2



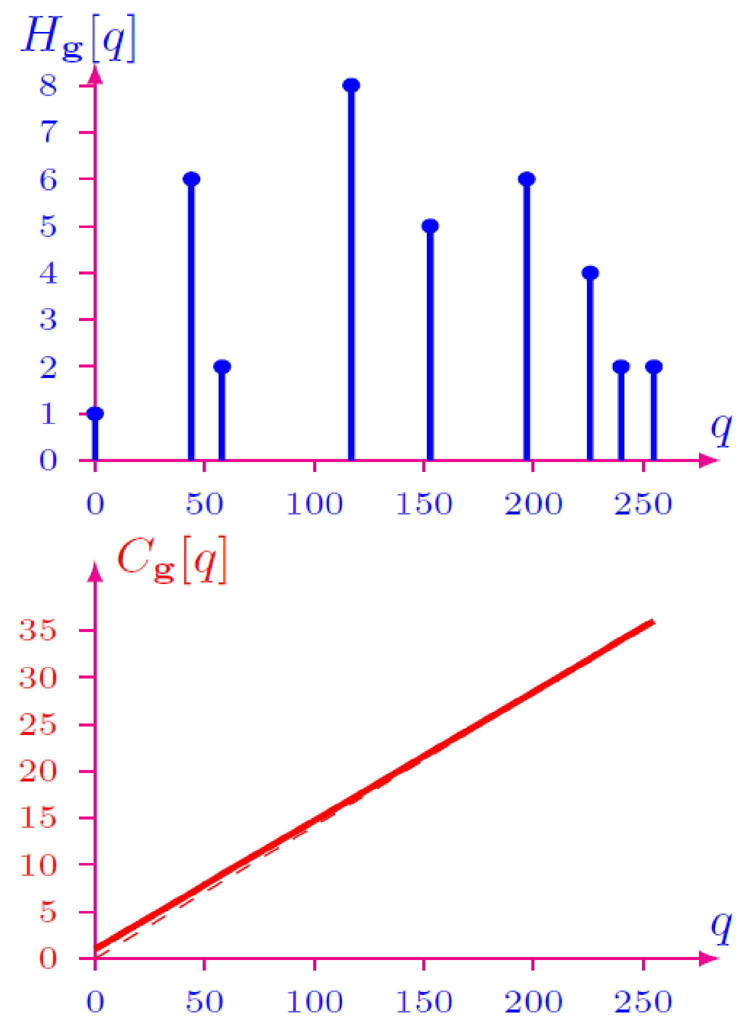
Histogram equalisation: 6x6 example

Image histograms (H) and cumulative histograms (C)

Before equalisation (image $\mathbf{f}$):

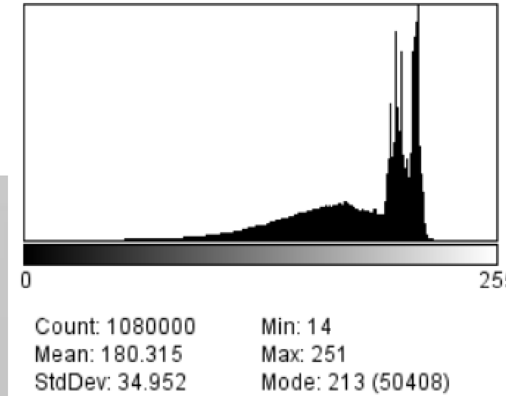


After equalisation (image $\mathbf{g}$):



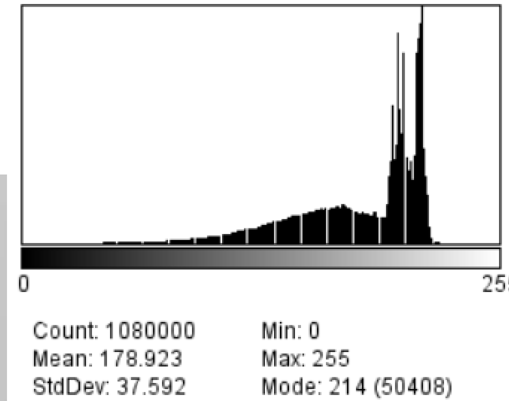
Summary


Original, low contrast


0
255
Count: 1080000
Mean: 180.315
StdDev: 34.952
Min: 14
Max: 251
Mode: 213 (50408)

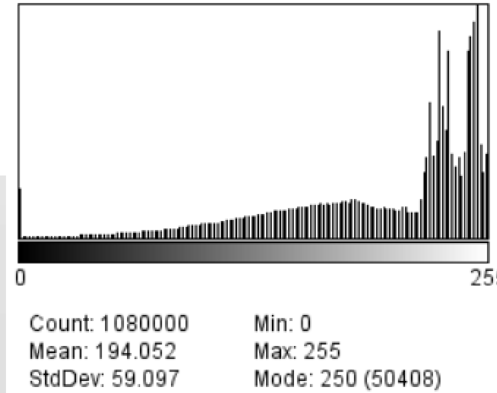
Summary

Contrast stretched



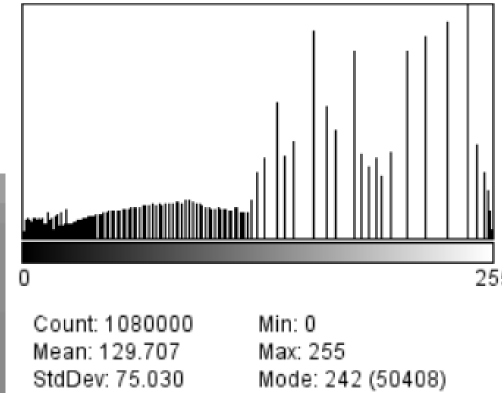
Summary


1-99 percentile contrast stretched

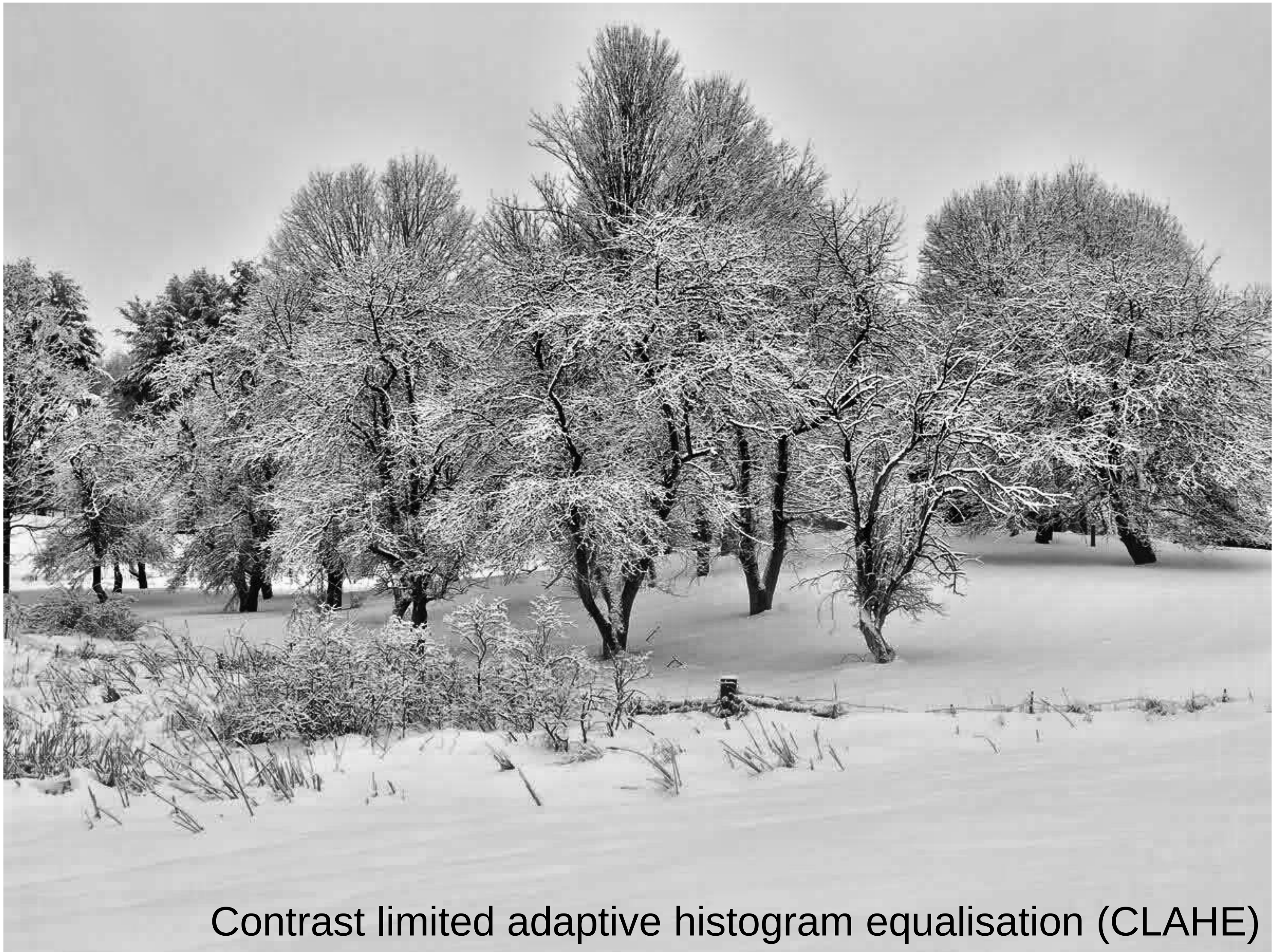


Summary

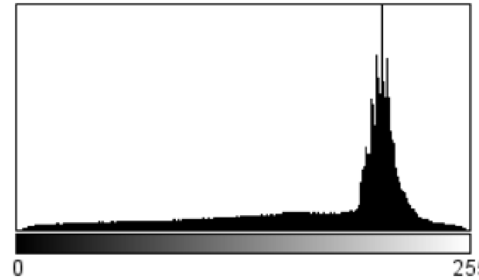

Histogram equalised


0
255
Count: 1080000
Mean: 129.707
StdDev: 75.030
Min: 0
Max: 255
Mode: 242 (50408)

Outlook state of the art



Contrast limited adaptive histogram equalisation (CLAHE)



Count: 1080000
Mean: 168.763
StdDev: 57.569

Min: 0
Max: 255
Mode: 206 (45837)

ImageJ:
blocksize: 127
histo bins: 256
max slope: 3.0

K. Zuiderveld: *Contrast Limited Adaptive Histogram Equalization*. In: P. Heckbert: *Graphics Gems IV*, Academic Press 1994, ISBN 0-12-336155-9