ECE 468 / CS 519:
Digital Image Processing

Histogram Equalization & Specification

Prof. Sinisa Todorovic

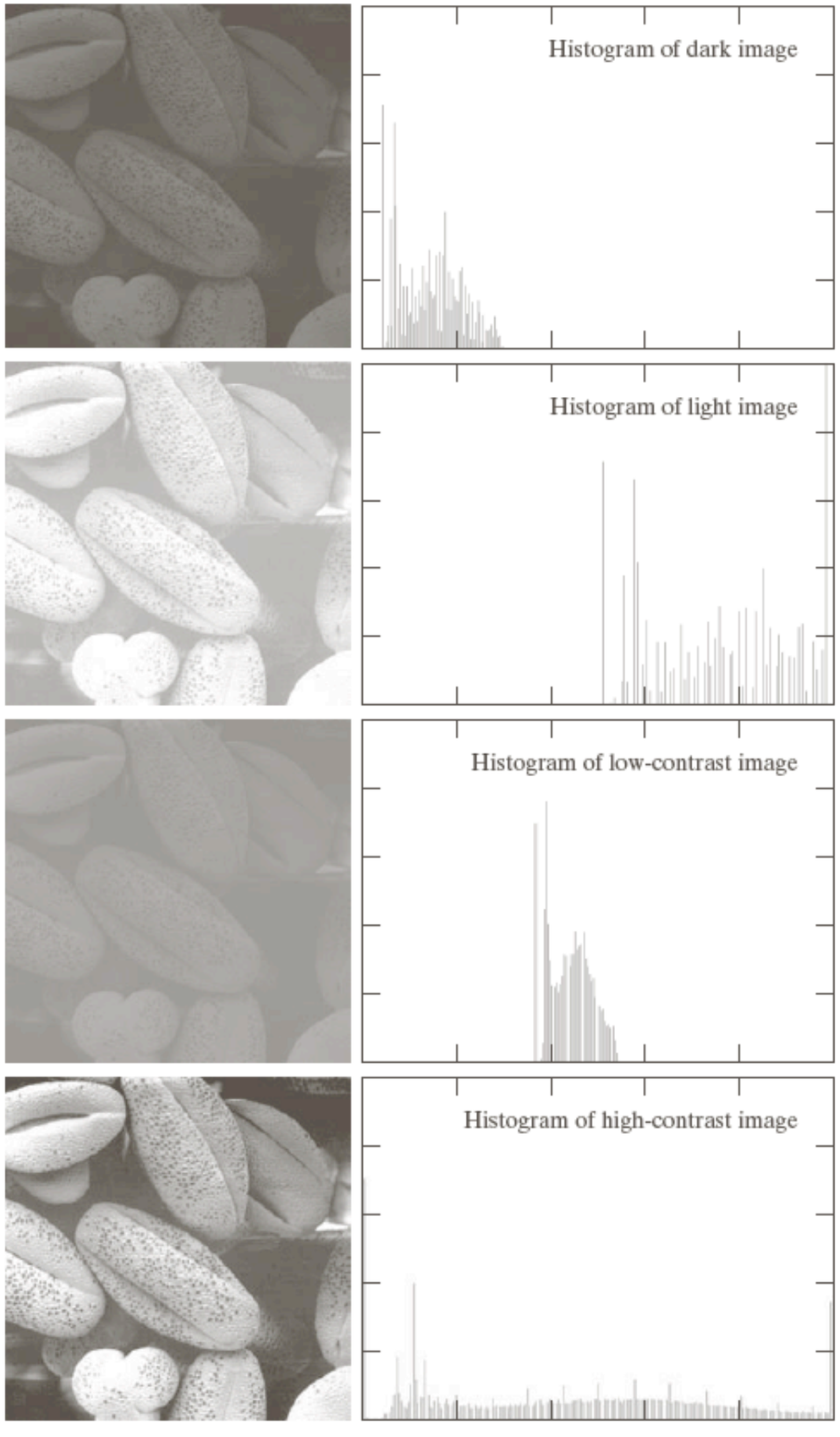
sinisa@eecs.oregonstate.edu

OSU Oregon State University

Outline

- Histogram equalization (Textbook: 3.3.1);
- Histogram specification (Textbook: 3.3.2);

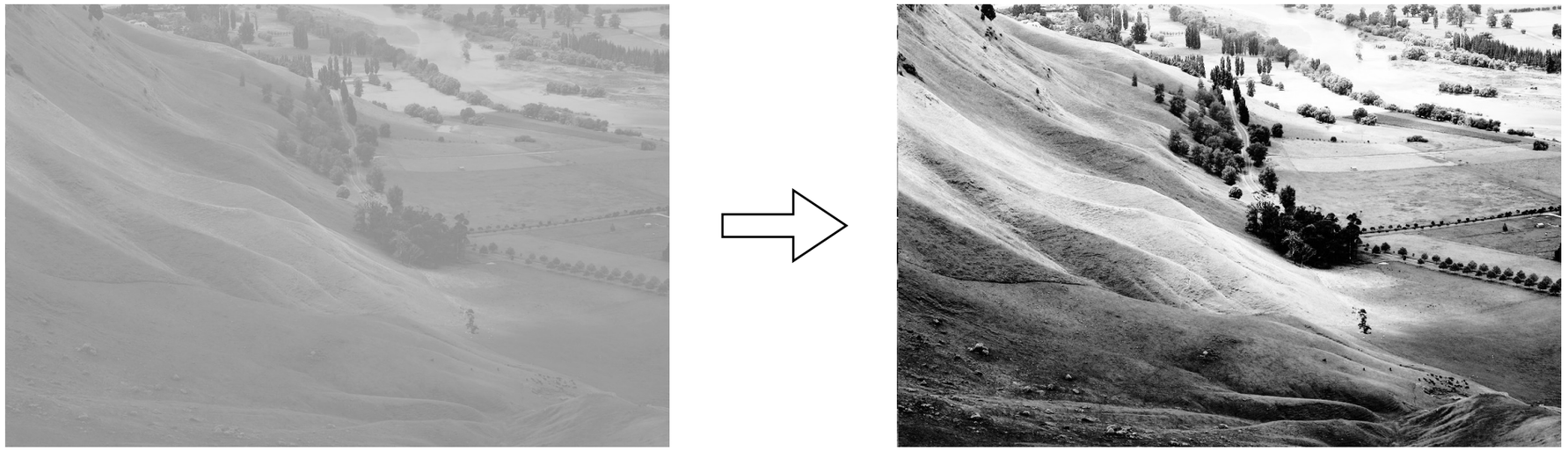
Histogram of Intensity Values



x axis: intensity values

y axis: frequency

Histogram Equalization



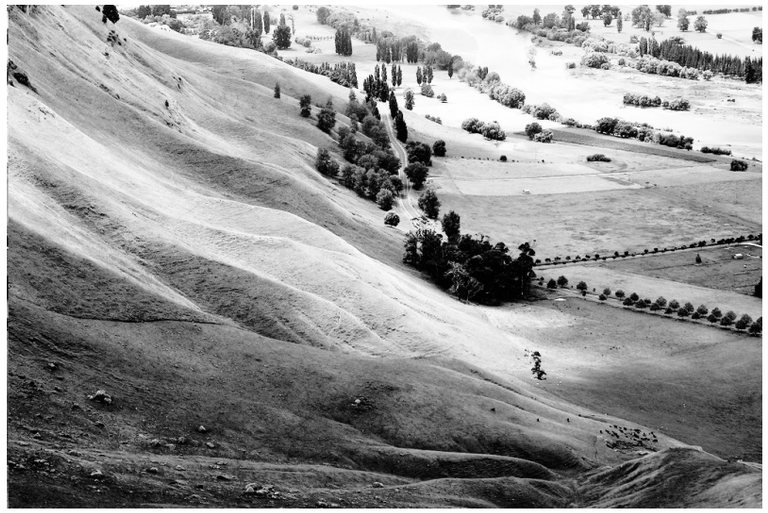
input

output

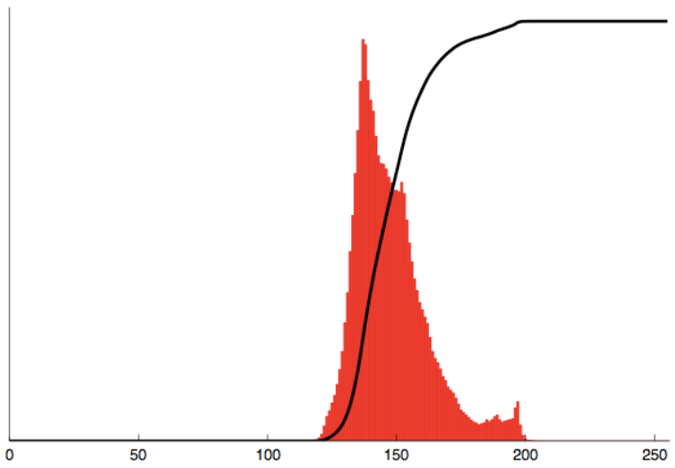
- Increases local contrast by spreading out the intensity histogram
- Produces artifacts

Histogram Equalization

input



output



intensity-level histogram (red)
cumulative histogram (black)

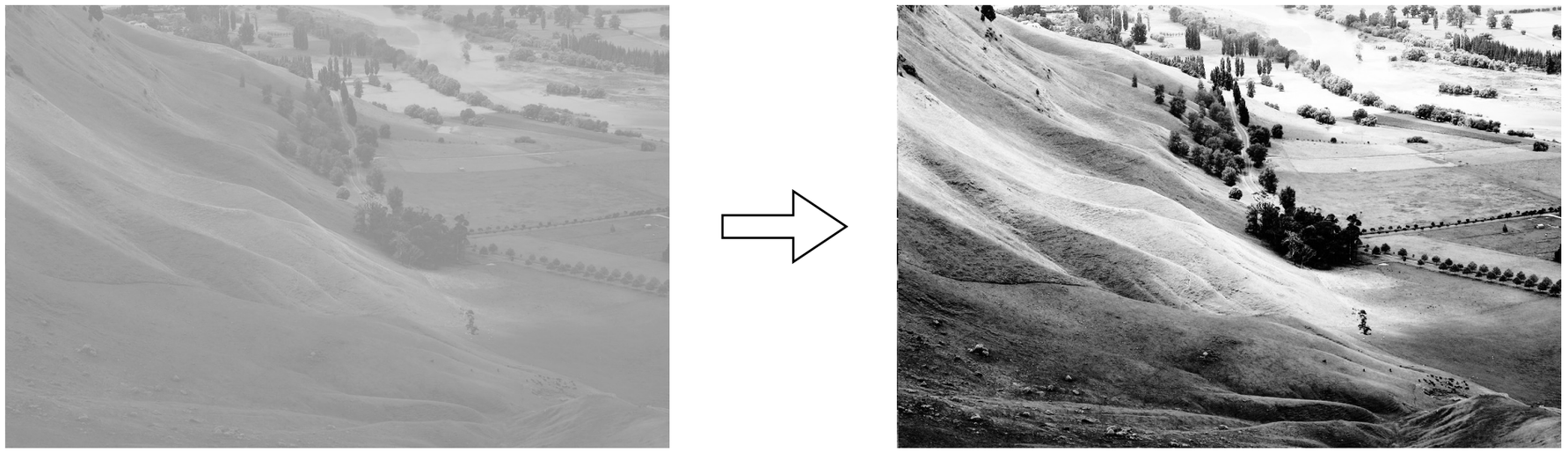
$$h(r_i \leq r < r_{i+1}) = \frac{n_i}{n}$$

number of pixels within bin *i*

bin *i*

total number of pixels

Histogram Equalization



input

output

$$T(r) = s$$

input intensity

output intensity

unequalized histogram of intensities

flat histogram of intensities

Transforming Density Functions

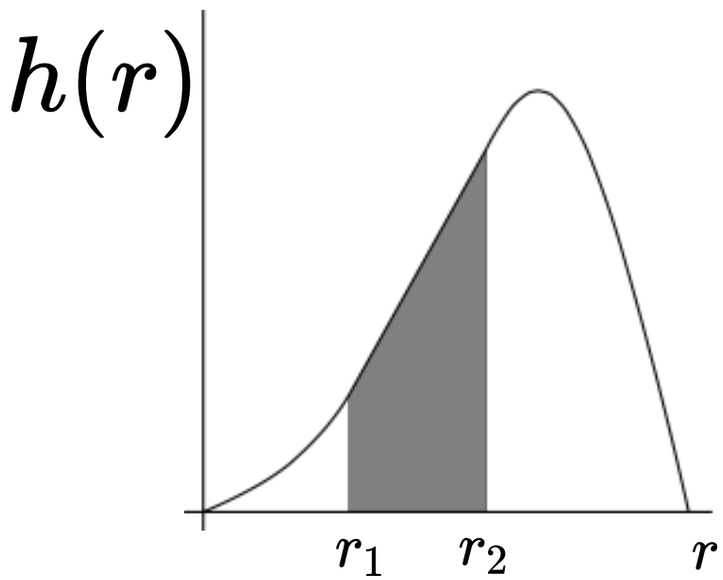
$$s = T(r)$$

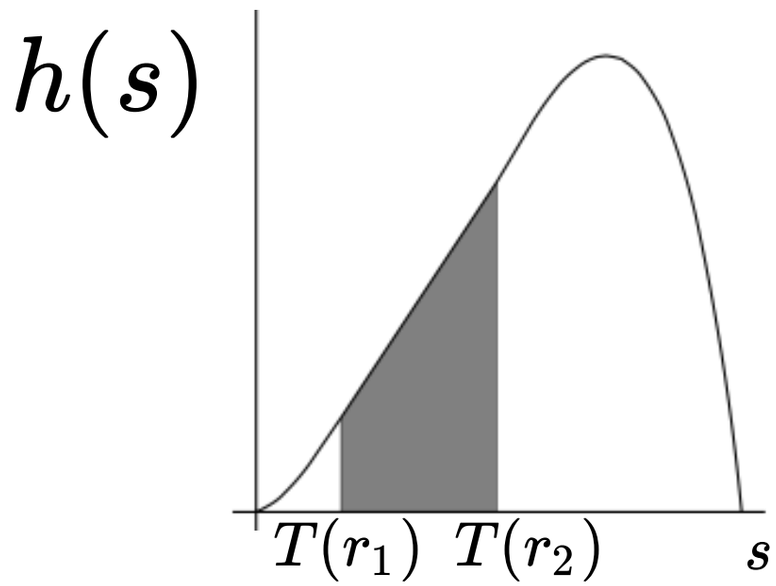
output pixel

input pixel

Key condition so as to keep the image content intact:

$$h(r_1 < r \leq r_2) = h(T(r_1) < s \leq T(r_2))$$





Transforming Density Functions

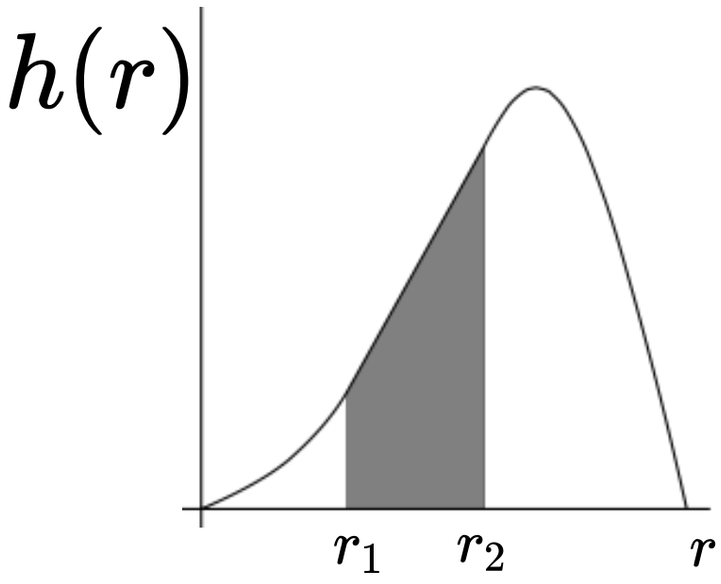
$$s = T(r)$$

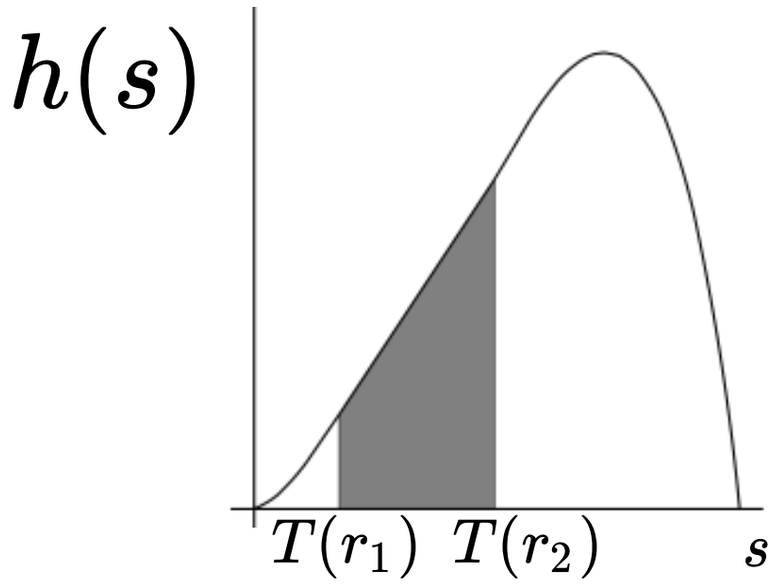
output pixel

input pixel

Key condition so as to keep the image content intact:

$$h(r)\Delta r = h(s)\Delta s$$





Transforming Density Functions

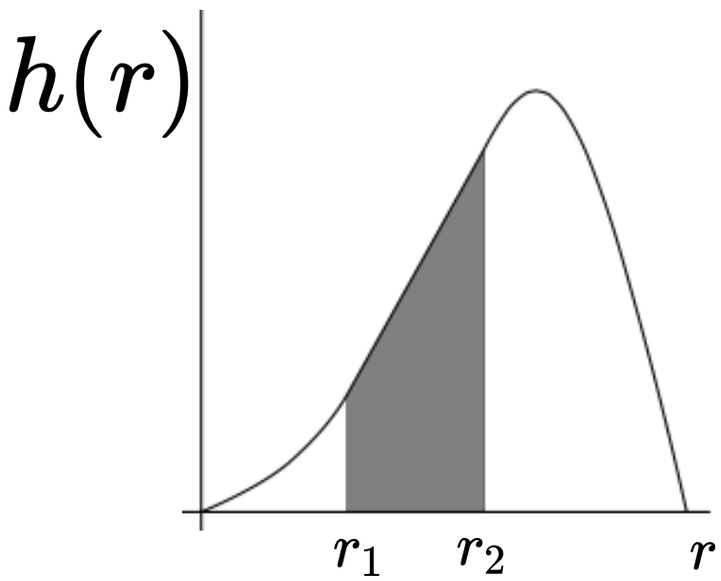
$$s = T(r)$$

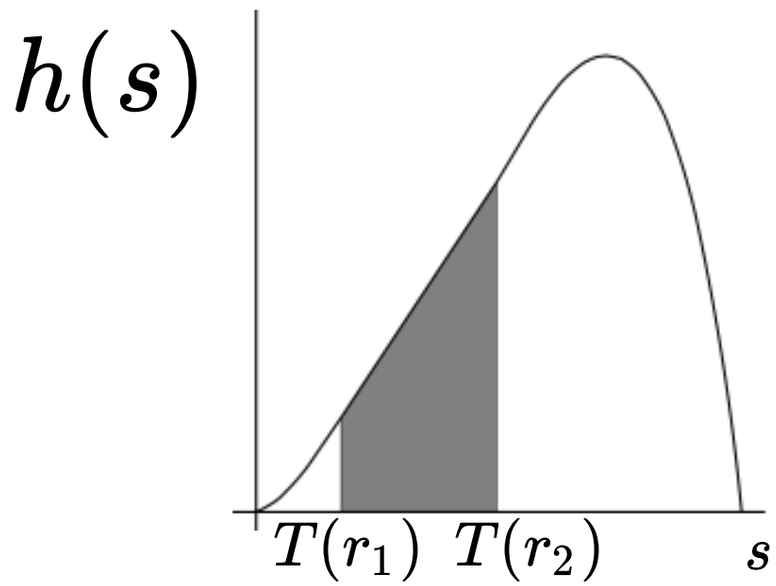
output pixel

input pixel

Key condition so as to keep the image content intact:

$$h(r)\Delta r = h(s)\Delta T(r)$$



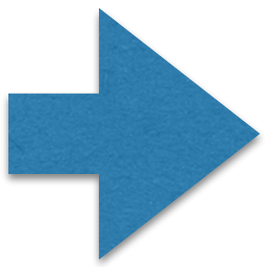


Transforming Density Functions

$$\overset{\text{output pixel}}{s} = T(\overset{\text{input pixel}}{r})$$

Key condition so as to keep the image content intact:

$$h(r)\Delta r = h(s)\Delta T(r)$$



$$\Delta T(r) = \frac{h(r)}{h(T(r))}\Delta r$$

Transforming Density Functions

$$s = T(r)$$

output pixel

input pixel

Key condition so as to keep the image content intact:

$$h(r)\Delta r = h(s)\Delta T(r)$$

$$\Delta T(r) = \frac{h(r)}{h(T(r))}\Delta r$$

$= 1$

Transforming Density Functions

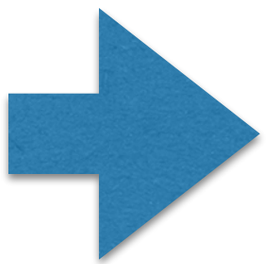
$$s = T(r)$$

output pixel

input pixel

Key condition so as to keep the image content intact:

$$h(r)\Delta r = h(s)\Delta T(r)$$



$$T(r) = \sum_{i=0}^{r} h(i)$$

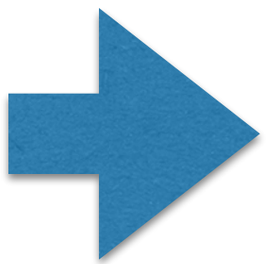
Transforming Density Functions

$$s = T(r)$$

output pixel: s; input pixel: r

Key condition so as to keep the image content intact:

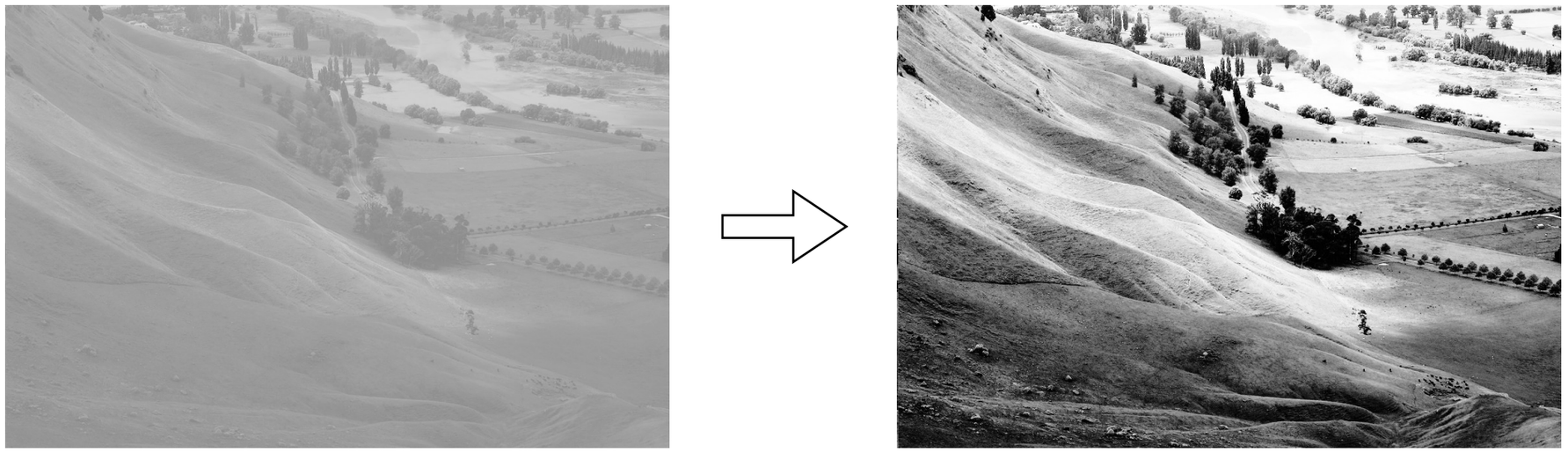
$$h(r)\Delta r = h(s)\Delta T(r)$$



$$s = T(r) = 255 \sum_{i=0}^{r} h(i)$$

for pixel values in [0,255]

Histogram Equalization

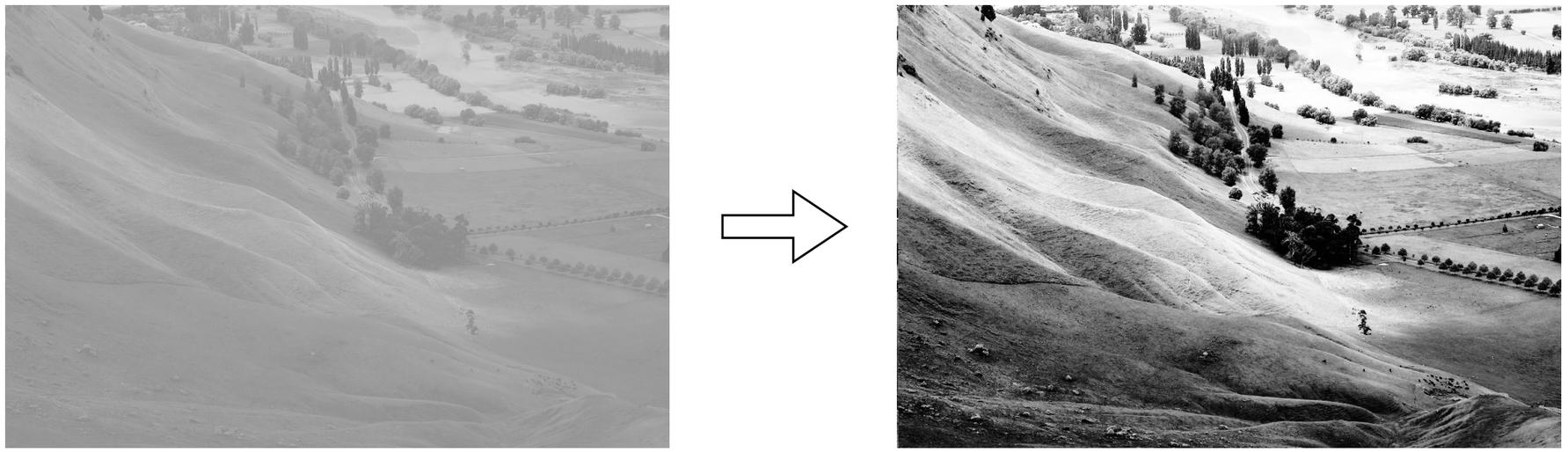


input output

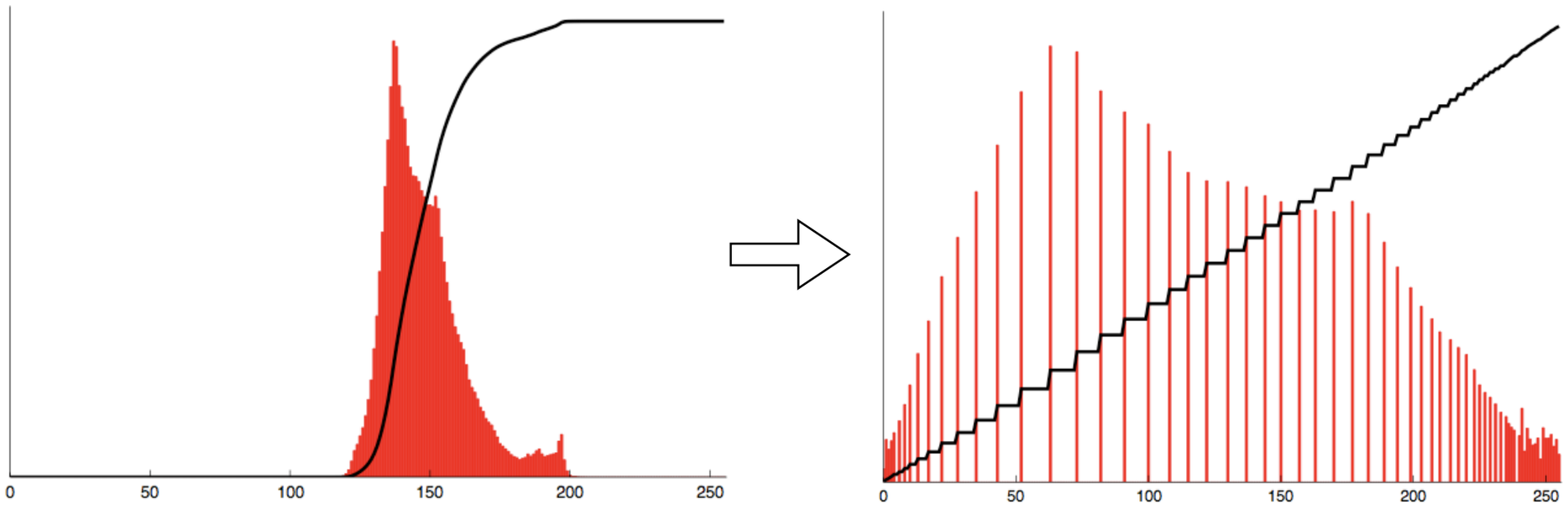
$$T(r) = c \sum_{i=0}^{r} h_{\text{input}}(i)$$

histogram

Histogram Equalization



input output

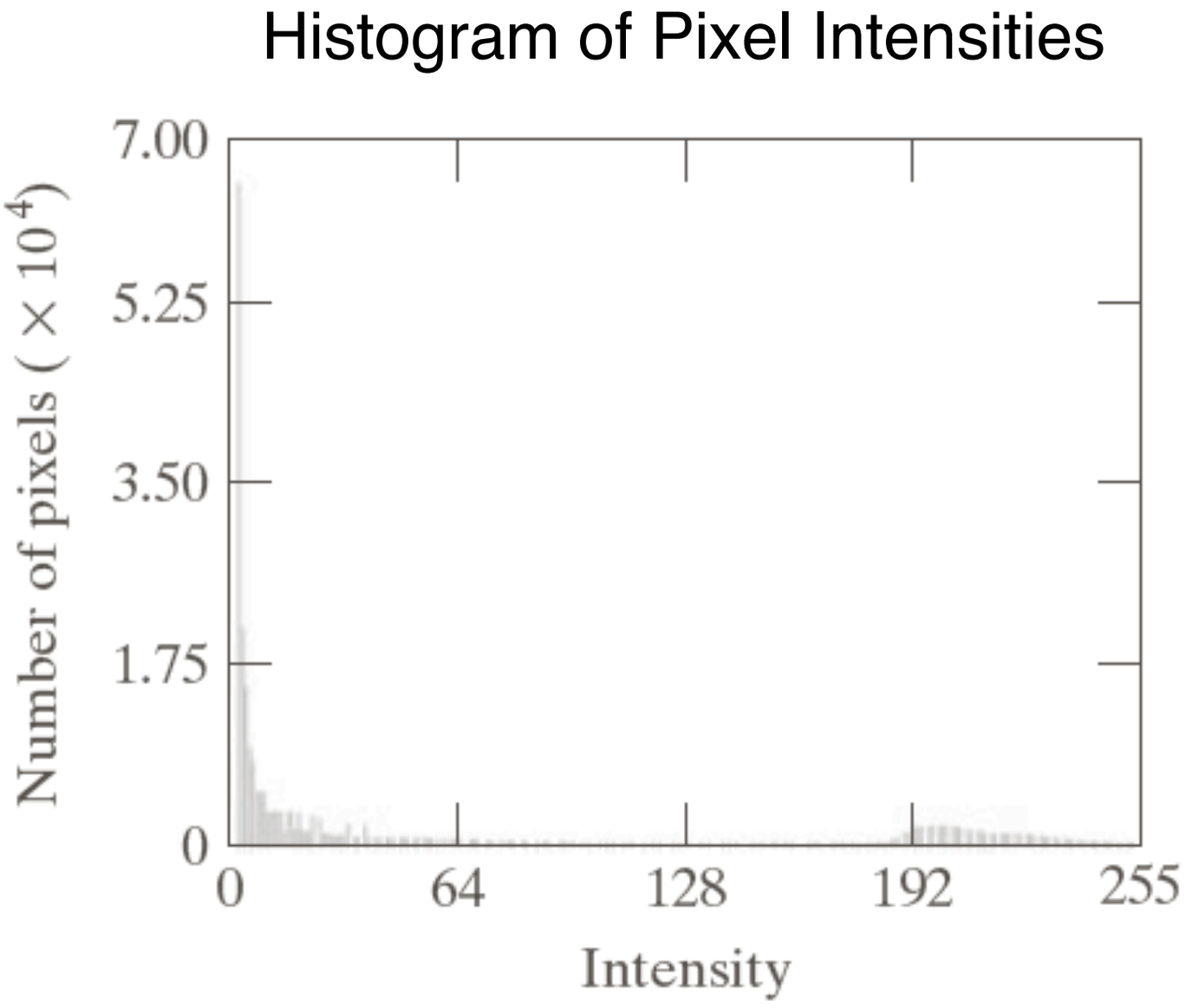


intensity-level histogram (red); cumulative histogram (black)

Example: Histogram Equalization

Example: Histogram Specification

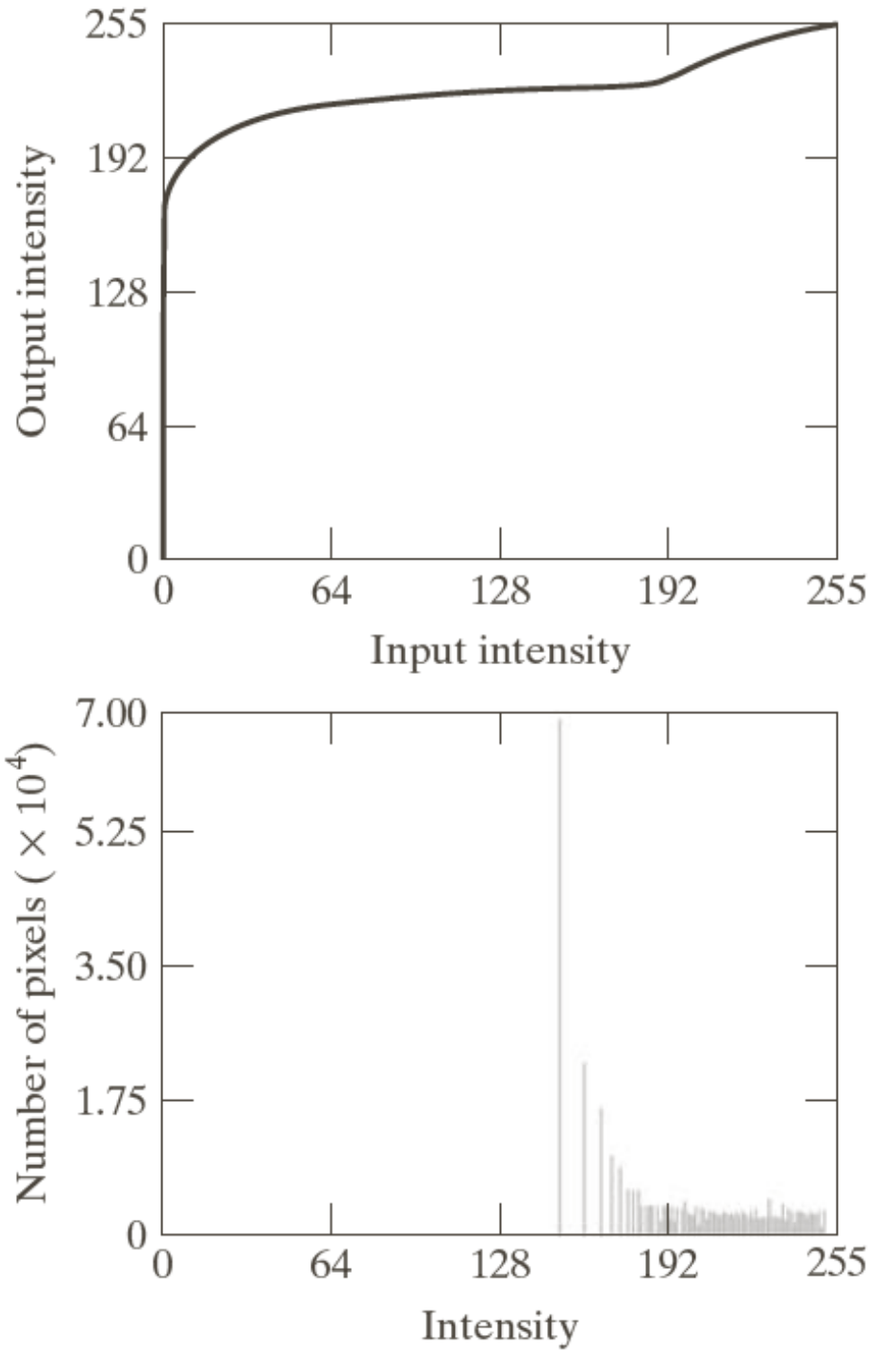
input image

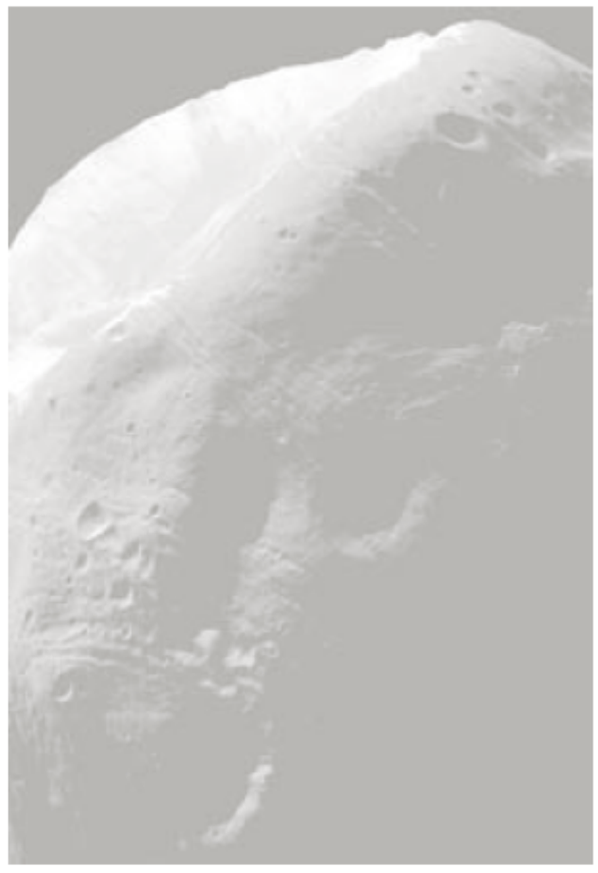


Example: Histogram Specification

Histogram Equalization

$$s = T(r)$$

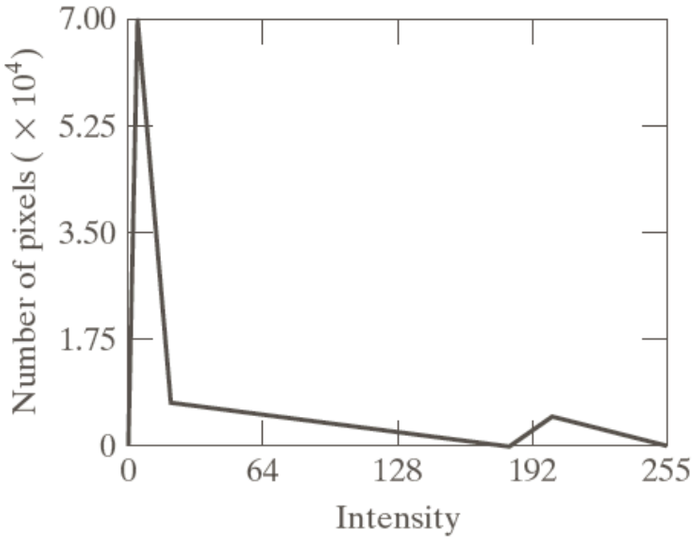




output

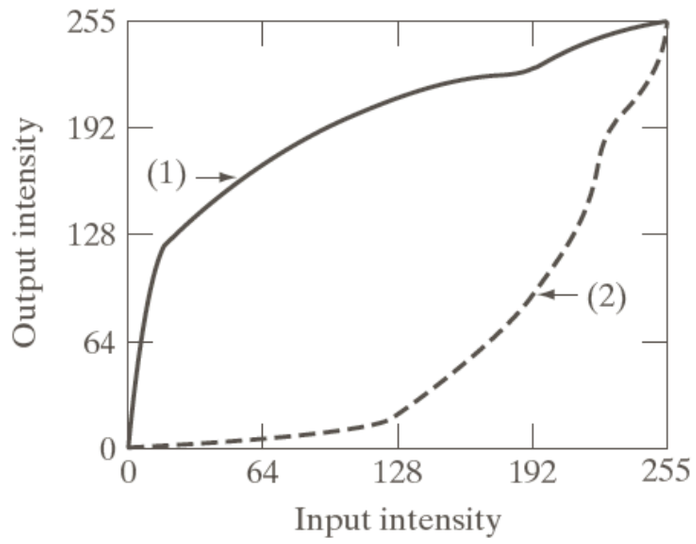
Example: Histogram Specification

specified histogram

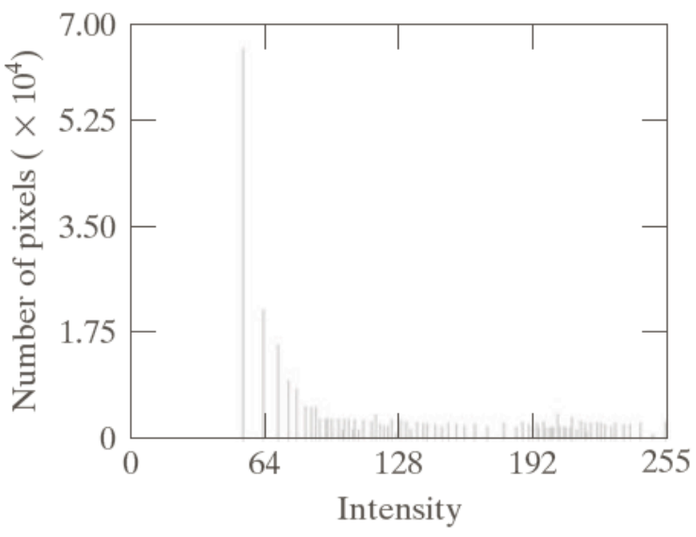


output

estimated transform *T(r)*



output histogram



Histogram Specification

$$s = T(r) = ?$$

Solving two equalization problems:

$$z = T_1(r)$$

equalize input r to z

$$z' = T_2(s)$$

equalize output s to z’

Histogram Specification

$$s = T(r) = ?$$

Solving two equalization problems:

$$z = T_1(r)$$

equalize input r to z

$$z' = T_2(s)$$

equalize output s to z’

it must be that: $z = z'$

Histogram Specification

$$s = T(r) = ?$$

Solving two equalization problems:

$$z = T_1(r)$$

equalize input r to z

$$z' = T_2(s)$$

equalize output s to z’

$$s = T_2^{-1}(T_1(r))$$

MATLAB

- `g = imadjust(f,[low_in; high_in],[low_out; high_out],gamma)`
  - **If high_out < low_out, the output is a photographic negative.**
- `h = imhist(f, num_bins);% histogram`
- `p = imhist(f, num_bins)/numel(f);% normalized histogram`
- `g = histeq(f, hspec),` **or** `g = histeq(f,256)`
  - **hspec - counts of equally spaced intensity values in [0,255]**

Next Class

- Spatial convolution and correlation (Textbook: 3.4.2);
- Smoothing and sharpening spatial filters (Textbook: 3.5)
- Matlab tutorial