SAIR

Spatial AI & Robotics Lab

CSE 473/573-A

L7: PYRAMIDS & HISTOGRAM

Chen Wang

Spatial AI & Robotics Lab

Department of Computer Science and Engineering

University at Buffalo The State University of New York

Content

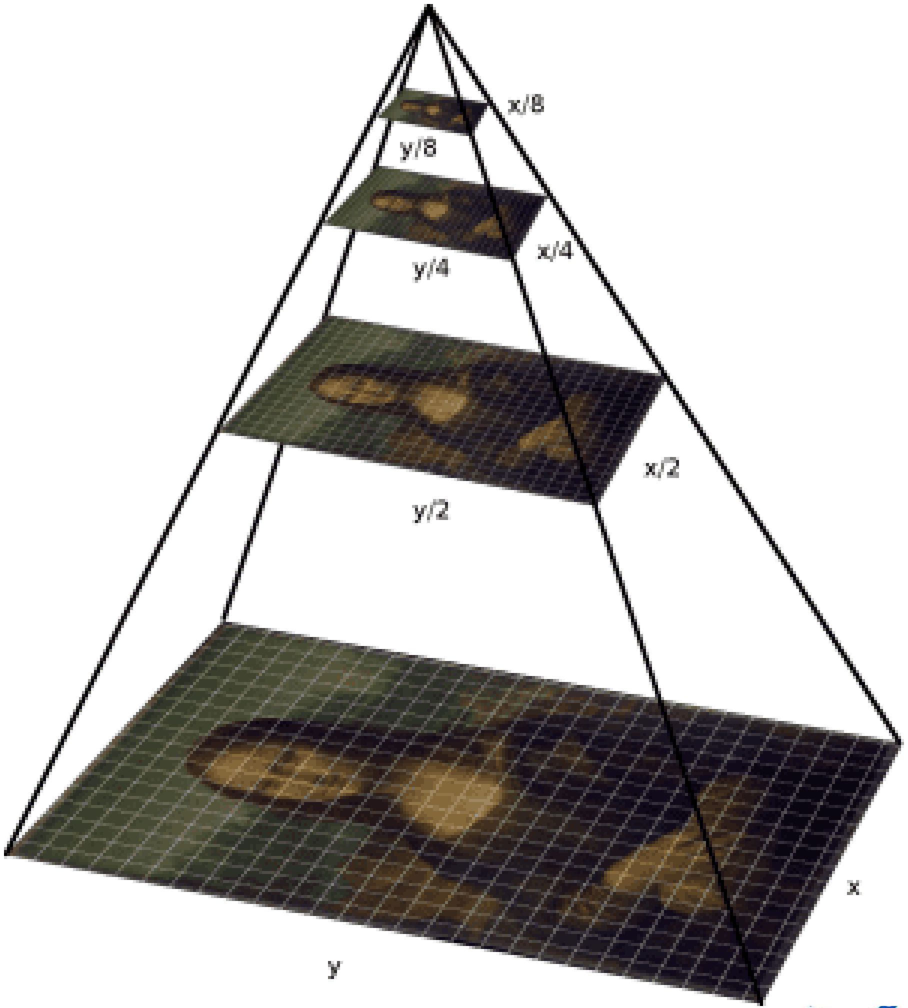
- Image Pyramids
  - Gaussian, Laplacian
  - Convolution and Transposed Convolution
- Image Histogram
  - Equalization, Matching
  - Image Enhancement
  - Histogram Equalization

IMAGE PROCESSING

Pyramids

Image Pyramids

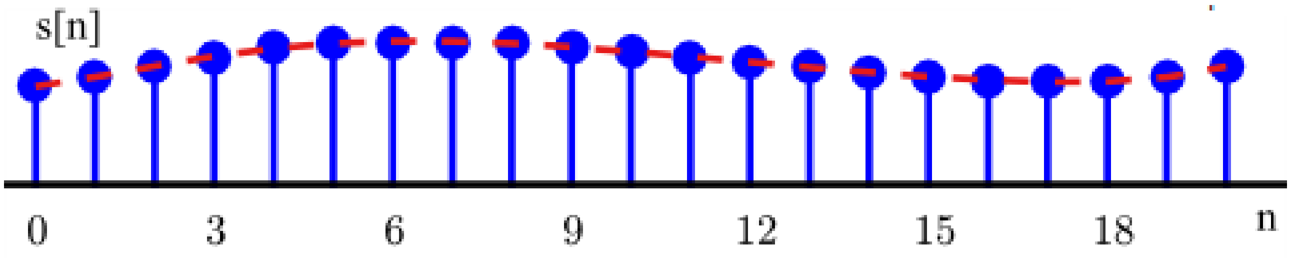
- Gaussian pyramid
- Laplacian pyramid
- Transposed Convolution

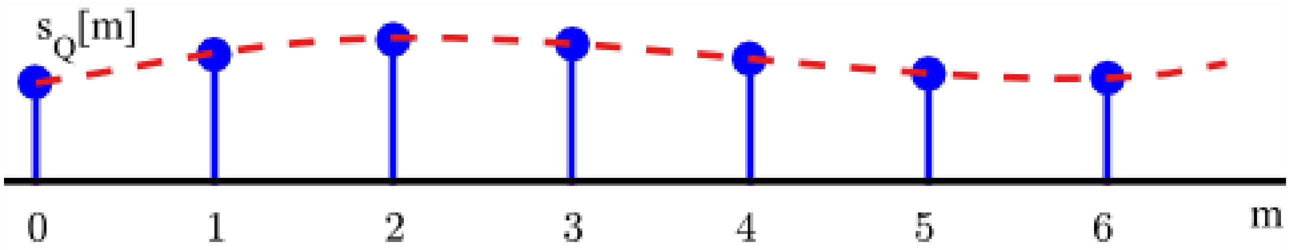


Pyramids applications

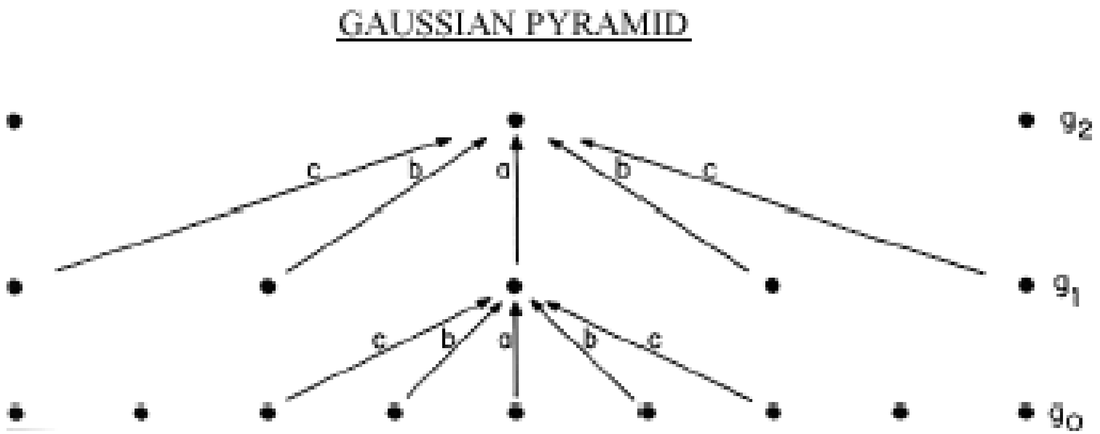
- Up- or Down- sampling images.
- Multi-resolution image analysis
  - Look for an object over various spatial scales
  - Coarse-to-fine image processing
    - form blur estimate or the motion analysis on very low-resolution image, up-sample and repeat.
  - Often a successful strategy for avoiding local minima in complicated estimation tasks.

Down-sampling





The Gaussian pyramid



g_0 = IMAGE

g_L = REDUCE [g_{L-1}]

Fig 1. A one-dimensional graphic representation of the process which generates a Gaussian pyramid Each row of dots represents nodes within a level of the pyramid. The value of each node in the zero level is just the gray level of a corresponding image pixel. The value of each node in a high level is the weighted average of node values in the next lower level. Note that node spacing doubles from level to level, while the same weighting pattern or "generating kernel" is used to generate all levels.

The Gaussian pyramid

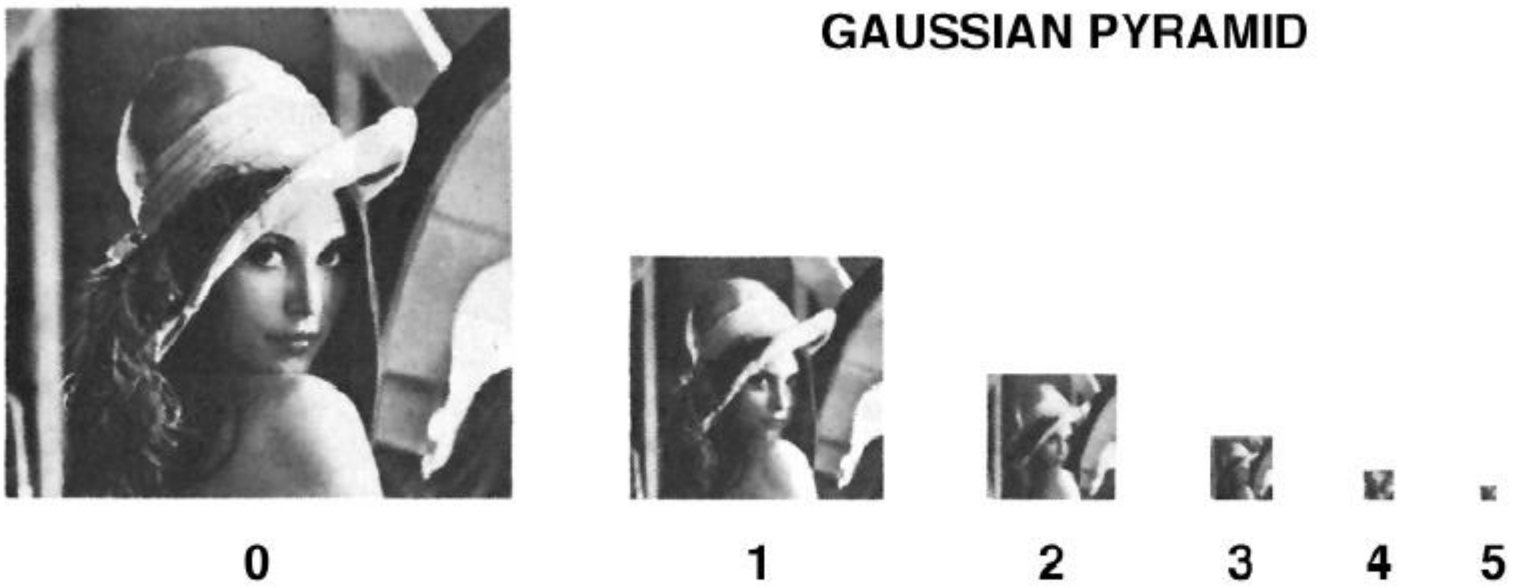


Fig. 4. First six levels of the Gaussian pyramid for the "Lady" image The original image, level 0, meusures 257 by 257 pixels and each higher level array is roughly half the dimensdons of its predecessor. Thus, level 5 measures just 9 by 9 pixels.

S A I R
Spatial AI & Robotics Lab

The Gaussian pyramid

512 256 128 64 32 16 8

Matrix operation for Down-sampling (1D)

- Assume x_1 is a signal with 16 elements.
- Down-sampled signal x_2 can be expressed as

$$x_2 = G_1 x_1$$

$$G_1 = \begin{matrix}
1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0
\end{matrix}$$

(Normalization constant of 1/16 omitted for visual clarity.)

Next pyramid level

$$x_3 = G_2 x_2$$

$$G_2 = \begin{matrix} 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 \end{matrix}$$

The combined effect of the two pyramid levels

- Smooth with Gaussians, because
  - a Gaussian * Gaussian = another Gaussian

$$x_3 = G_2 G_1 x_1$$

$$G_2 G_1 =$$

1	4	10	20	31	40	44	40	31	20	10	4	1	0	0	0	0	0	0	0
0	0	0	0	1	4	10	20	31	40	44	40	31	20	10	4	1	0	0	0
0	0	0	0	0	0	0	0	1	4	10	20	31	40	44	40	30	16	4	0
0	0	0	0	0	0	0	0	0	0	0	0	1	4	10	20	25	16	4	0

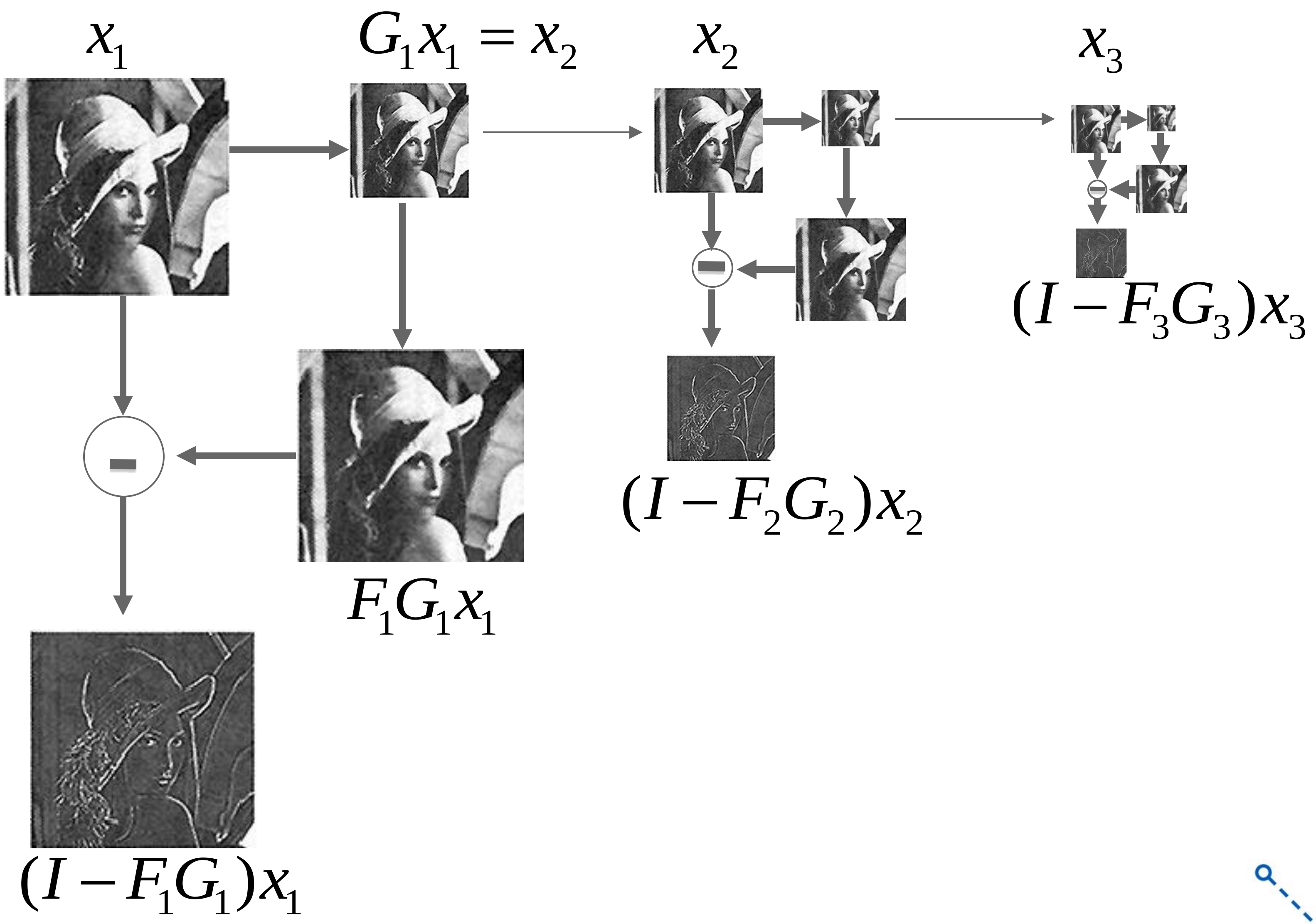
Up-sampling

$$y_2 = F_3 x_3 \qquad F_3 = \begin{matrix} 6 & 1 & 0 & 0 \\ 4 & 4 & 0 & 0 \\ 1 & 6 & 1 & 0 \\ 0 & 4 & 4 & 0 \\ 0 & 1 & 6 & 1 \\ 0 & 0 & 4 & 4 \\ 0 & 0 & 1 & 6 \\ 0 & 0 & 0 & 4 \end{matrix}$$

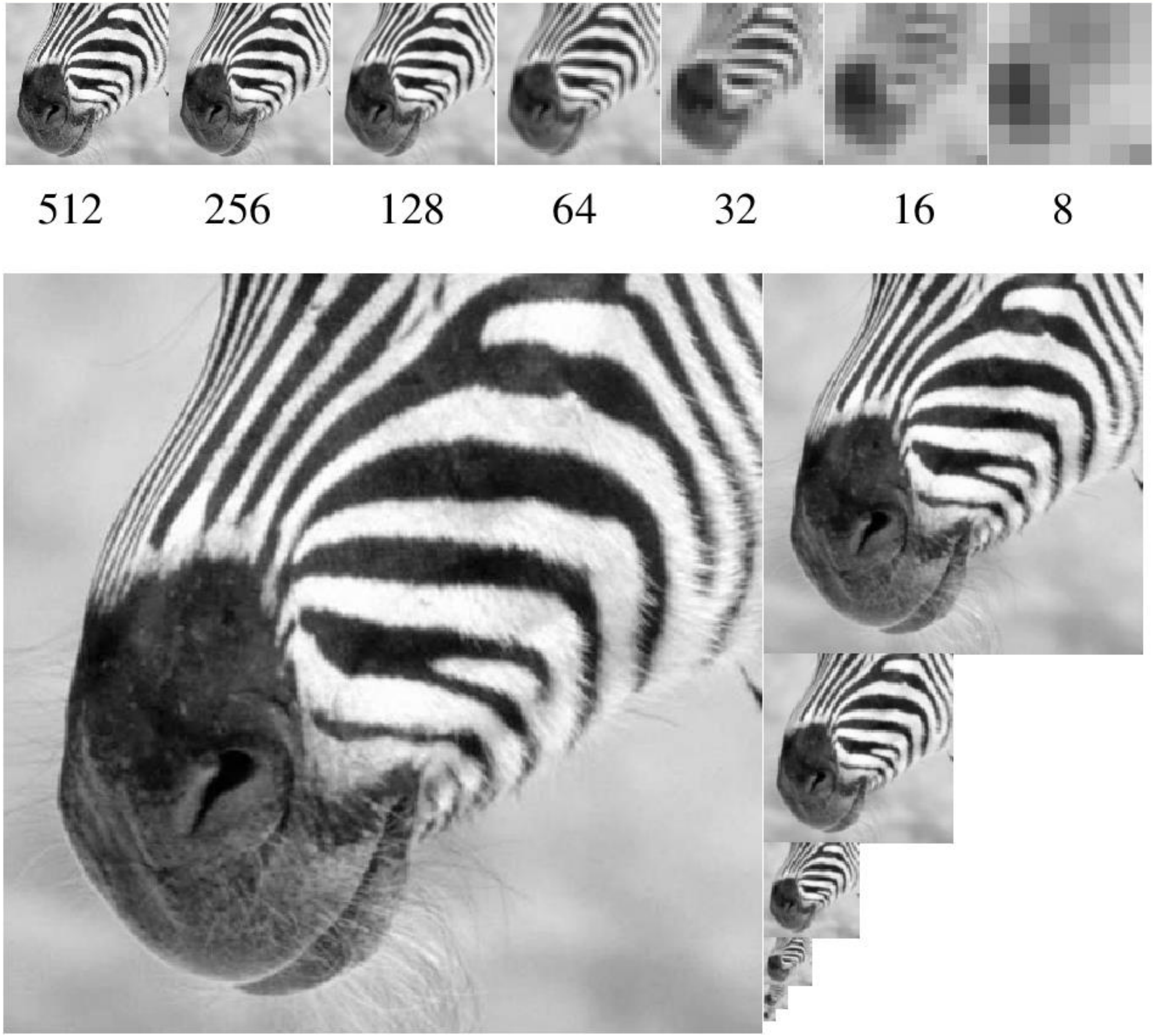
The Laplacian Pyramid

- **Difference between up-sampled Gaussian pyramid and Gaussian pyramid**.
- **Band pass filter** - each level represents spatial frequencies (largely) unrepresented at other level.

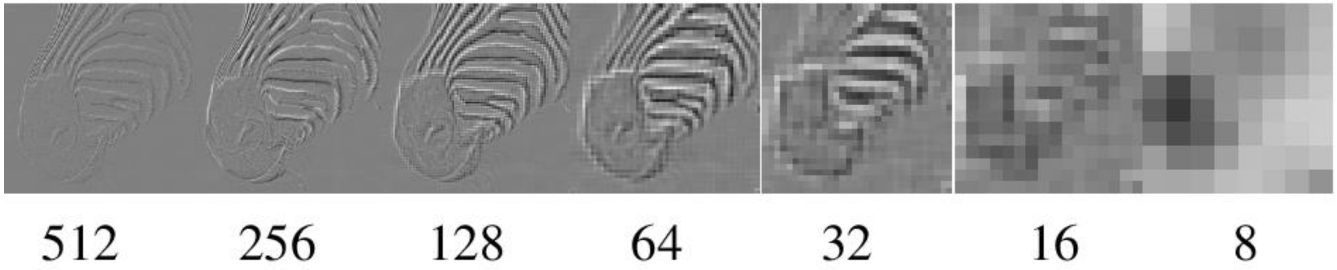
Laplacian pyramid algorithm

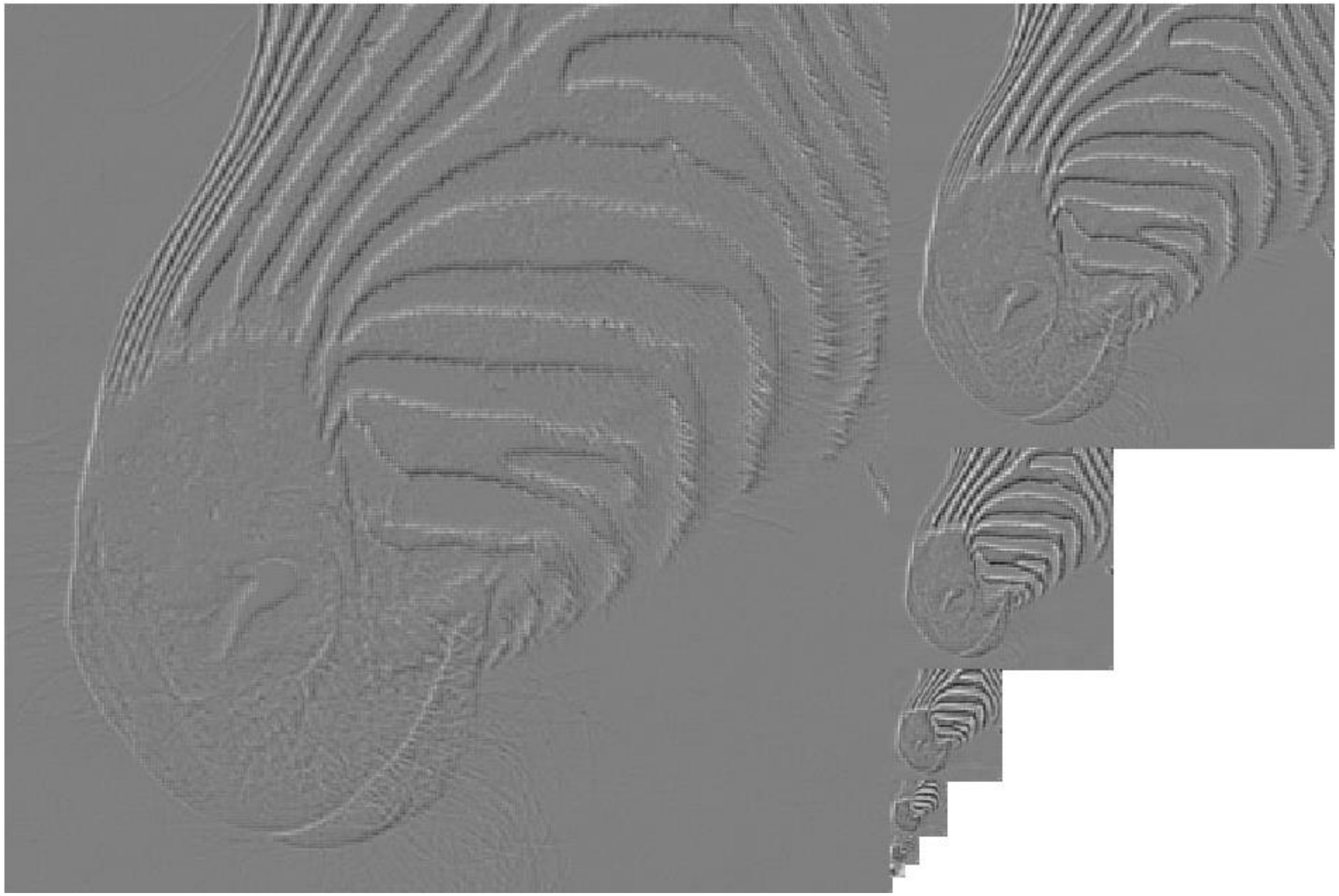


Gaussian pyramid



Laplacian pyramid





SAIR
Spatial AI & Robotics Lab

Laplacian Pyramid

- Information captured at each level of a Gaussian (top) and Laplacian (bottom) pyramid
  - showing full resolution.

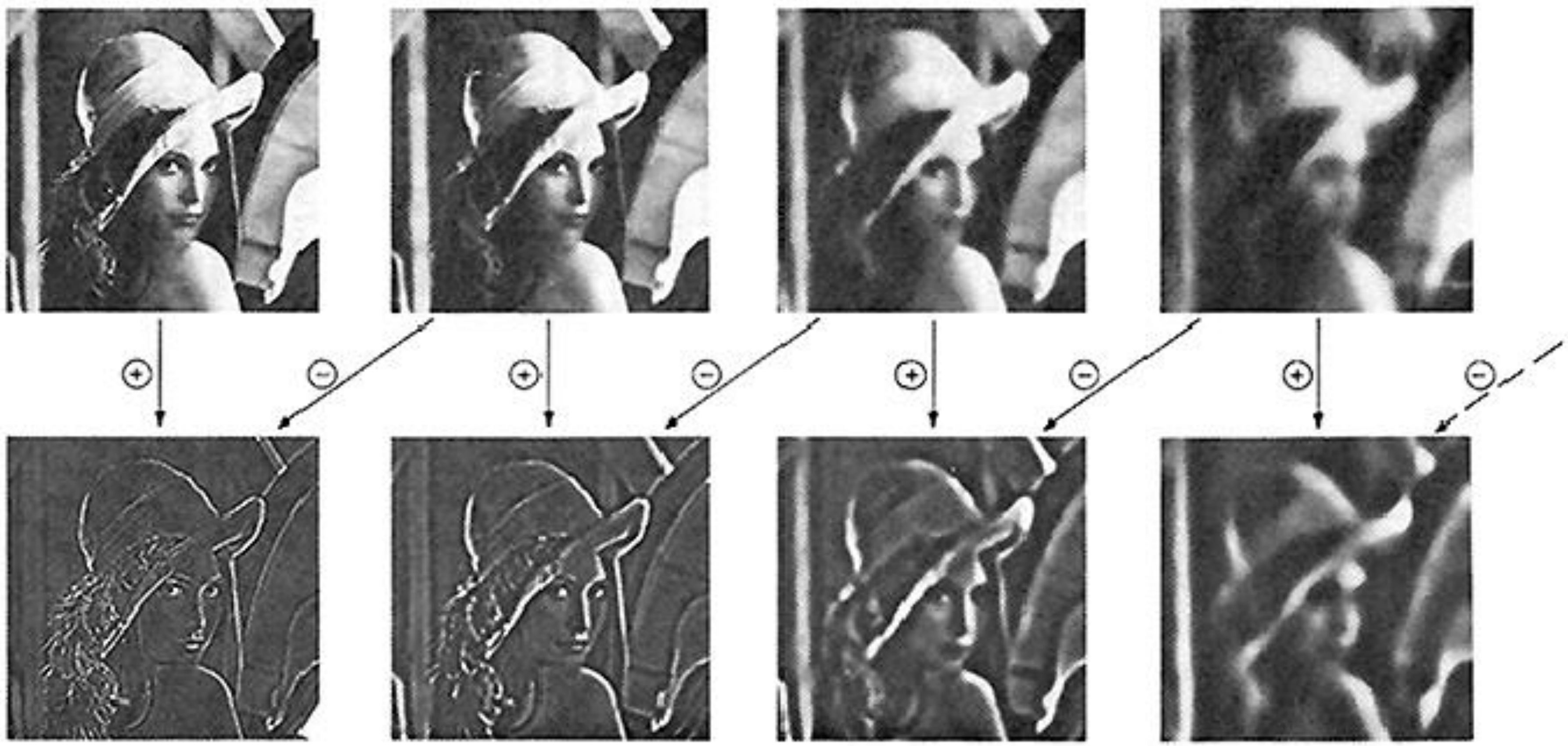
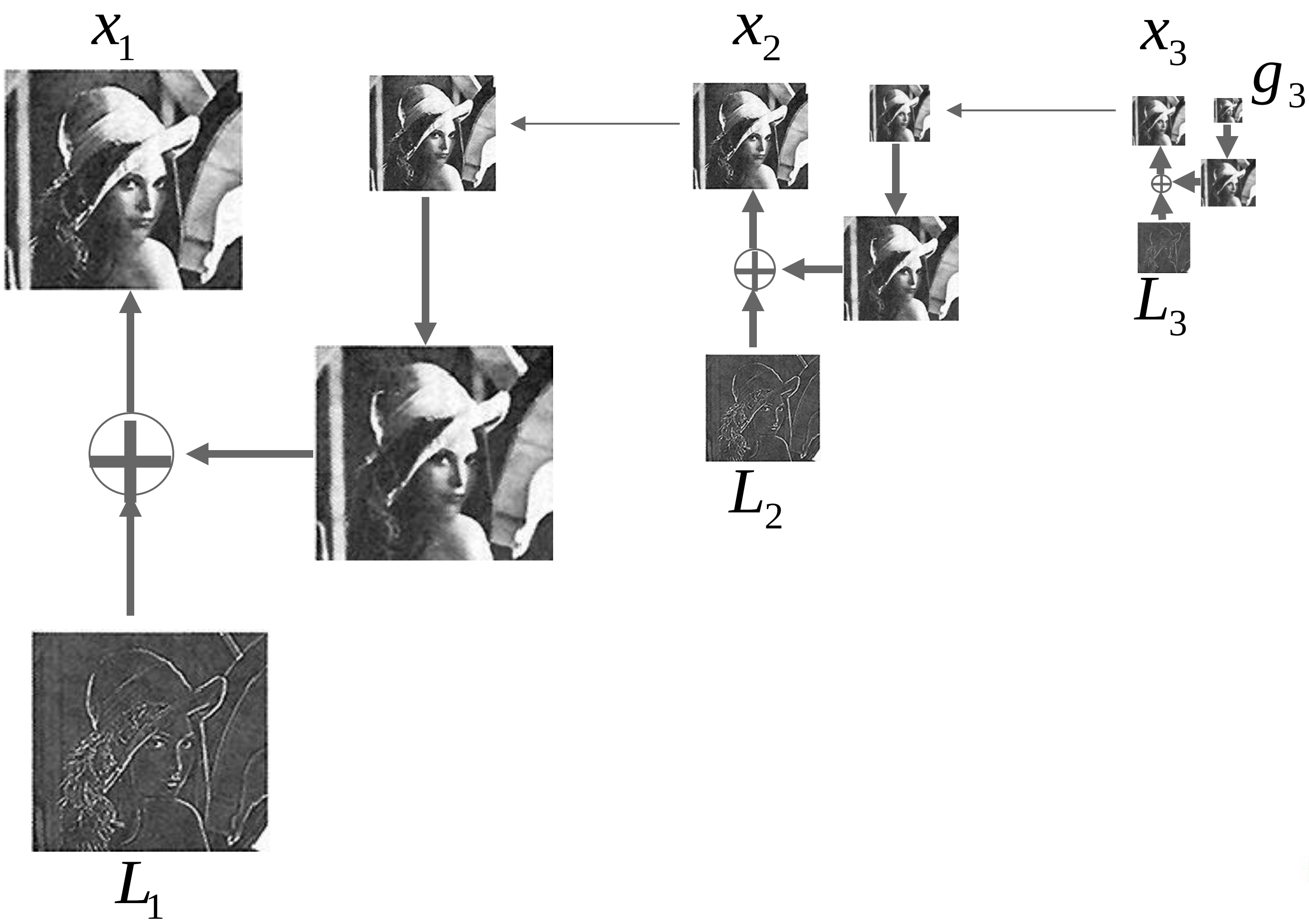


Fig 5. First four levels of the Gaussian and Laplacian pyramid. Gaussian images, upper row, were obtained by expanding pyramid arrays (Fig. 4) through Gaussian interpolation. Each level of the Laplacian pyramid is the difference between the corresponding and next higher levels of the

Laplacian Pyramid

- Reconstruction: recover x_1 from L_1, L_2, L_3 and g_3



Laplacian Pyramid Reconstruction algorithm

G# is the blur-and-downsample operator at pyramid level #
F# is the blur-and-upsample operator at pyramid level #

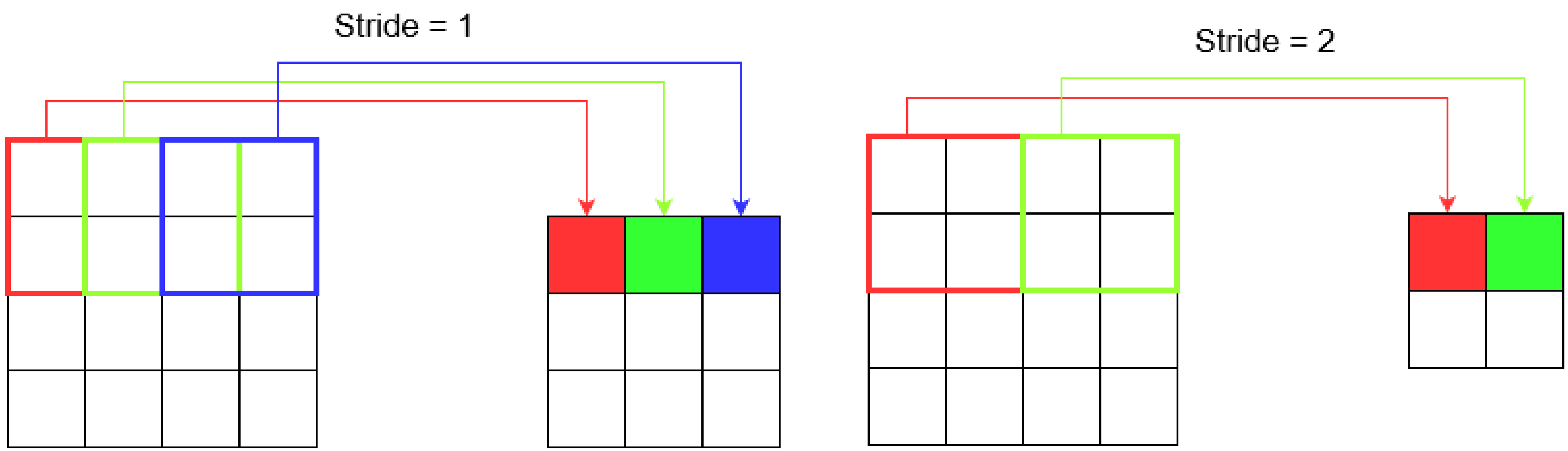
Laplacian pyramid elements:

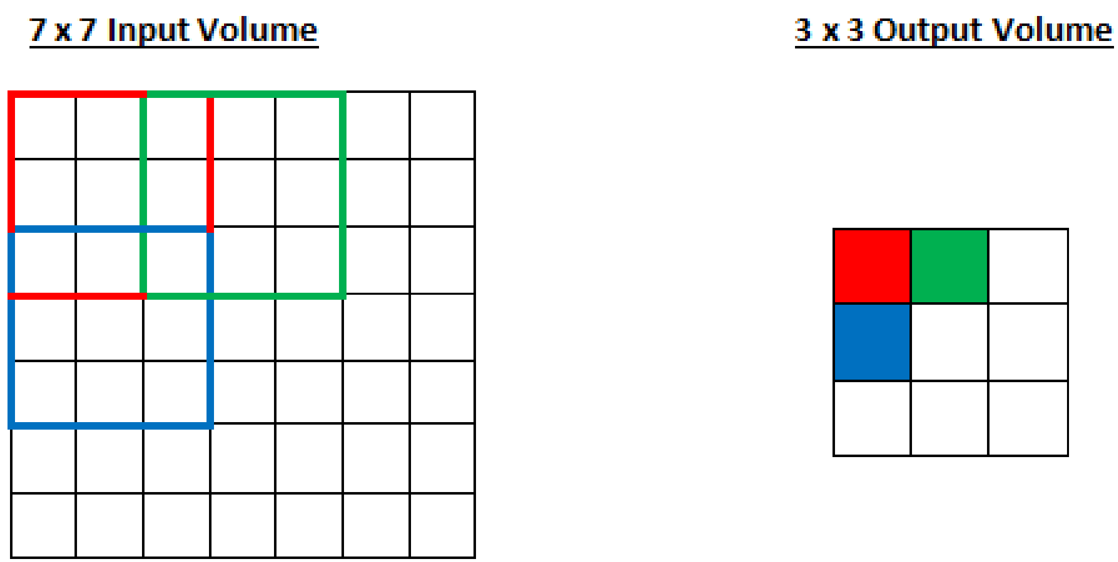
$$L1 = (I - F1\ G1)\ x1$$
$$x2 = G1\ x1$$
$$L2 = (I - F2\ G2)\ x2$$
$$x3 = G2\ x2$$
$$L3 = (I - F3\ G3)\ x3$$
$$x4 = G3\ x3$$

Reconstruction of original image (x1) from Laplacian pyramid elements:

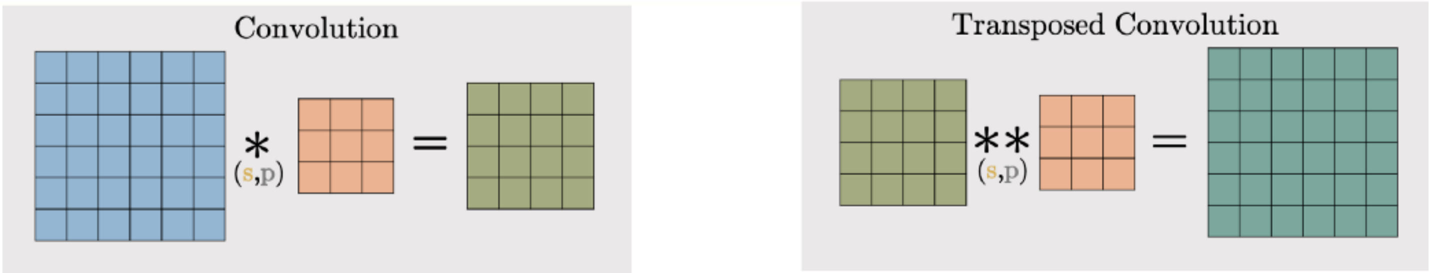
$$x3 = L3 + F3\ x4$$
$$x2 = L2 + F2\ x3$$
$$x1 = L1 + F1\ x2$$

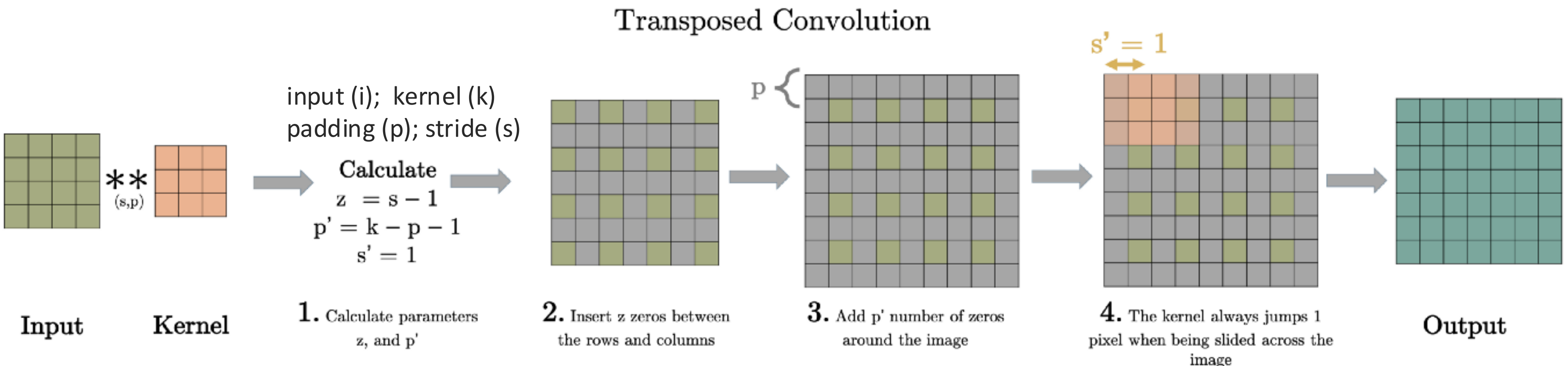
Convolution for Down-sampling

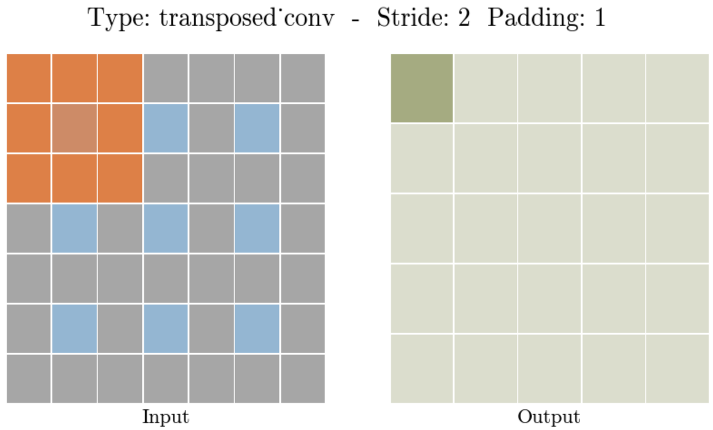




Transposed Convolution for Up-sampling








```
import torch, torch.nn.functional as F

inputs = torch.randn(2, 4, 5, 5)
kernel = torch.randn(4, 8, 3, 3)
F.conv_transpose2d(inputs, kernel, stride=2, padding=1)
```

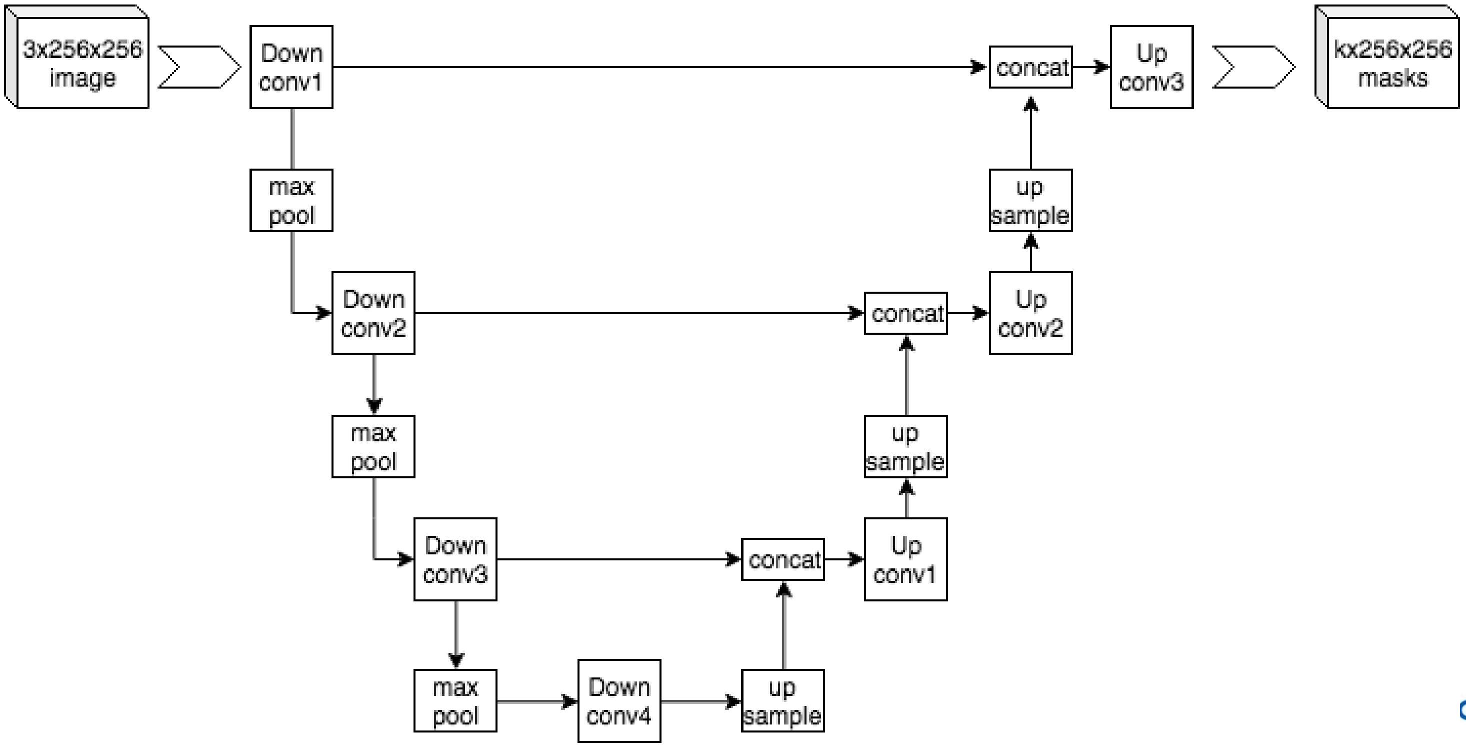
Guess the shape of output. (2, 8, 9, 9)

SAIR
Spatial AI & Robotics Lab

https://towardsdatascience.com/what-is-transposed-convolutional-layer-40e5e6e31c11

Why use these representations?

- Handle real-world size variations
- Remove noise, Analyze texture
- Recognize objects, Label image features



Shelhamer E, Long J, Darrell T (April 2017). "Fully Convolutional Networks for Semantic Segmentation". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **39** (4): 640–651

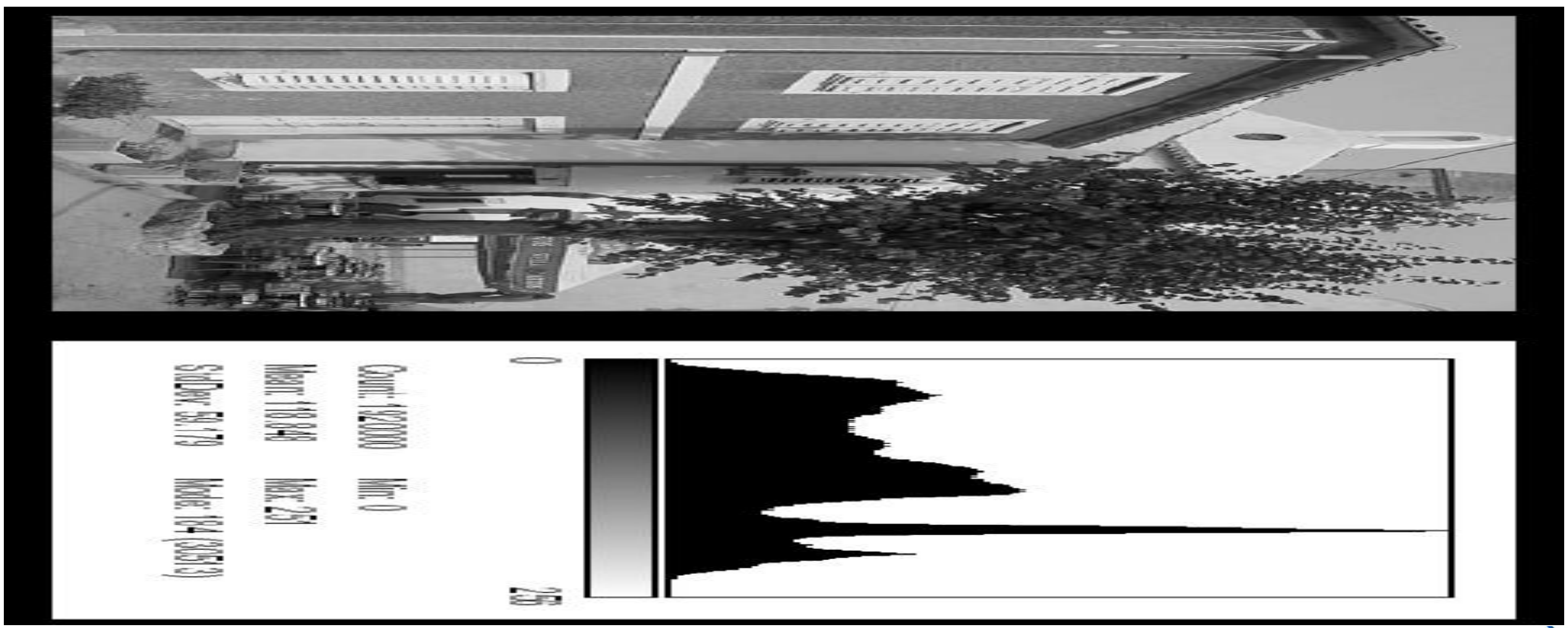
SAIR
Spatial AI & Robotics Lab

IMAGE PROCESSING

Histogram

Histogram

- Histograms plots how many times (frequency) each intensity value in image occurs
  - Image (left) has 256 distinct gray levels (8 bits)
  - Histogram (right) shows frequency (how many times) each gray level occurs



Histogram

- Many cameras display real time histograms of scene
  - Helps avoid taking over-exposed pictures

Histogram

- A histogram for a grayscale image with intensity values in range

$$I(u, v) \in [0, K-1]$$

  - would contain exactly K entries
- For 8-bit grayscale image, K = 2^8 = 256
- Each histogram entry is defined as:
  - h(i) = number of pixels with intensity i (0<i<K).
  - h(255) = number of pixels with intensity i = 255.

Formal definition

$$\mathsf{h}(i) = \mathrm{card}\big\{(u, v) \mid I(u, v) = i\,\big\}$$

Number (size of set) of pixels

such that

Normalized Histogram

Histogram $h(r_k) = n_k$

r_k is the k^{th} intensity value

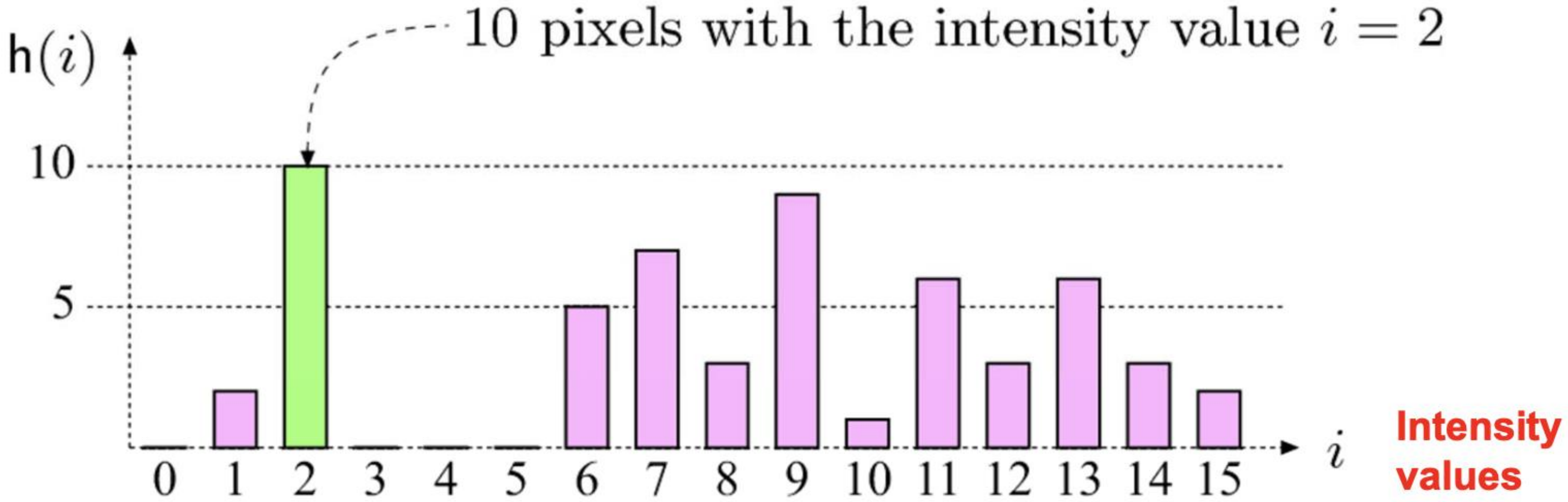
n_k is the number of pixels in the image with intensity r_k

Normalized histogram $p(r_k) = \dfrac{n_k}{MN}$

n_k: the number of pixels in the image of size $\mathrm{M} \times \mathrm{N}$ with intensity r_k

Probability density function

Histogram

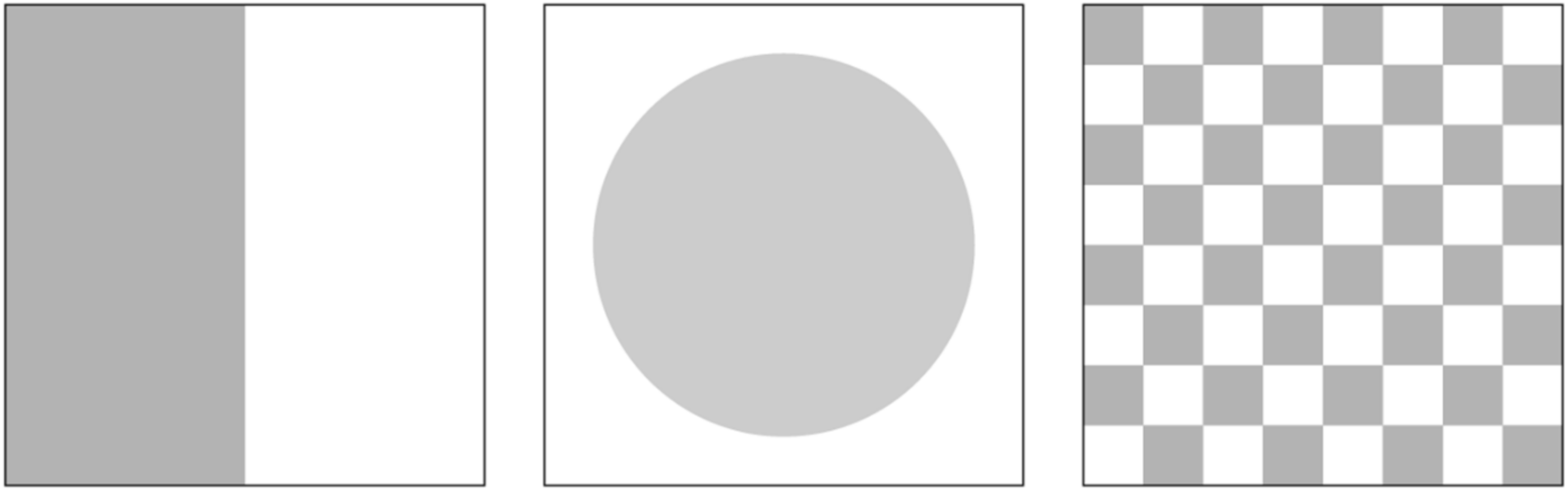


h(i)	0	2	10	0	0	0	5	7	3	9	1	6	3	6	3	2
i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

- E.g., K = 16, 10 pixels have intensity value = 2
- Histograms: only statistical information
- No indication of location of pixels

Histogram

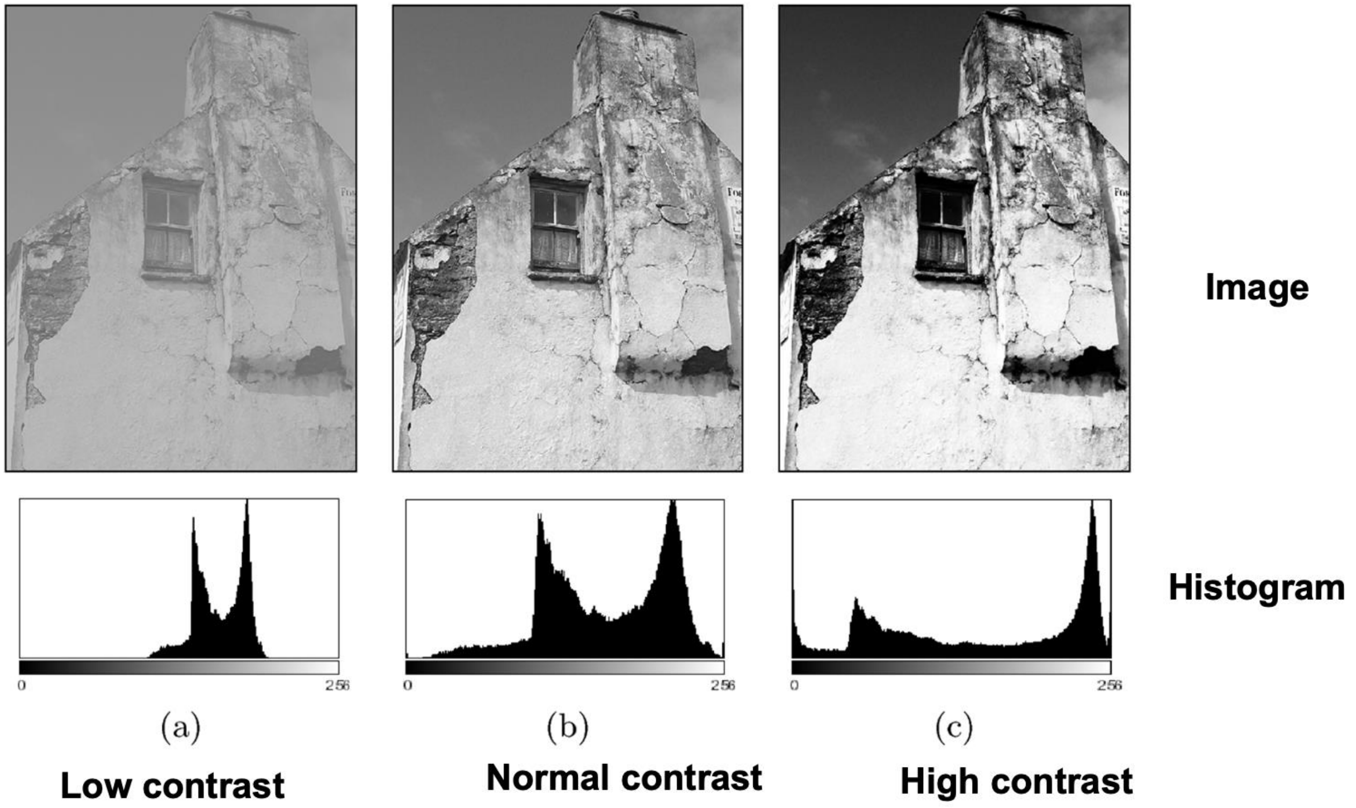
- Different images can have same histogram
- 3 images below have same histogram



- Half of pixels are gray, half are white
  - Same histogram = same statistics
  - Distribution of intensities could be different
- Can we reconstruct image from histogram? No!

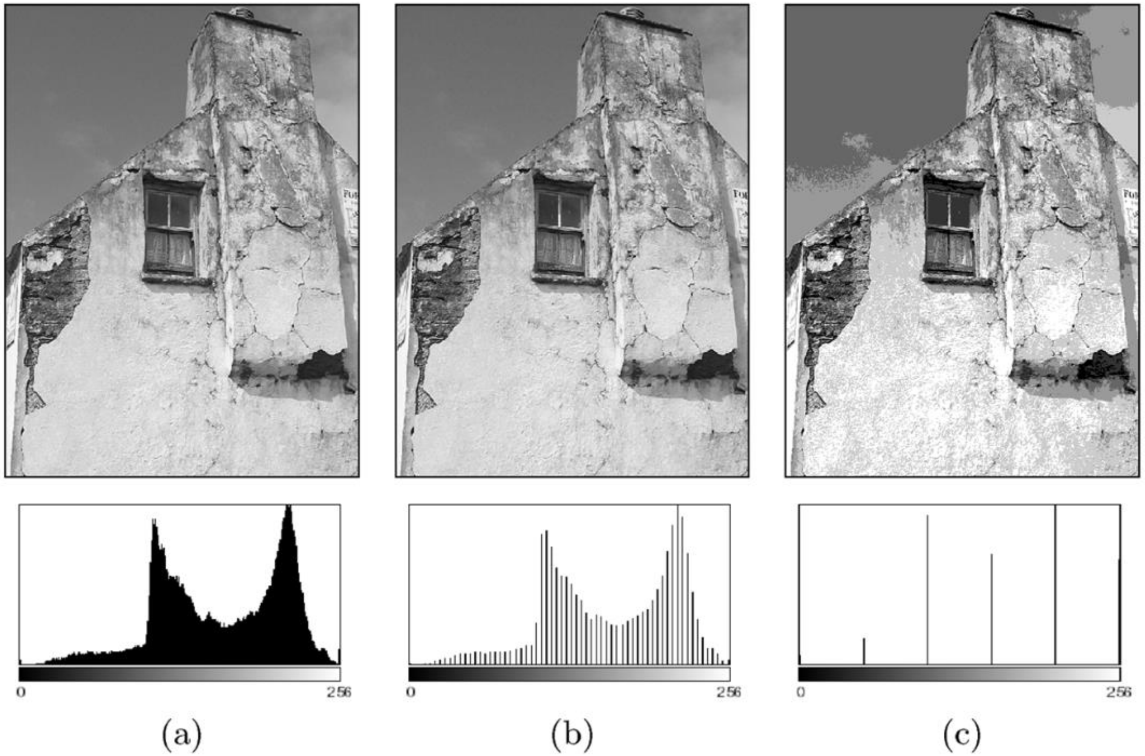
Histogram and Contrast

- What is Good Contrast?
  - Widely spread intensity values
  - Large difference between min and max intensity



Histogram and Dynamic Range

- Dynamic Range: number of distinct pixels in image
  - High dynamic range means very bright and very dark parts in a single image (many distinct values)



High Dynamic Range

Low Dynamic Range (64 intensities)

Extremely low Dynamic Range (6 intensity values)

Large Histograms: Binning

- High resolution image can yield very large histogram
- E.g., 32-bit image = 2^{32} = 4,294,967,296 columns
  - Such a large histogram impractical to display
- Solution is Binning
  - Combine ranges of values into histogram columns

So, given the image $I : \Omega \rightarrow [0, K-1]$, the binned histogram for I is the function

$$h(i) = \mathrm{card}\{(u, v) |\ a_i \leq I(u, v) < a_{i+1}\},$$

where $0 = a_0 < a_1 < \ldots < a_B = K$.

Number (size of set) of pixels

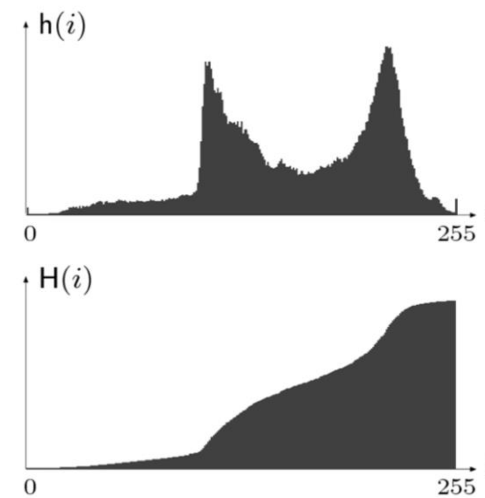
such that

Pixel's intensity is between a_i and a_{i+1}

Cumulative Histogram

- Useful for histogram equalization (introducing later)
- Similar to the Cumulative Density Function (CDF)
- Definition

$$\mathsf{H}(i) = \sum_{j=0}^{i} \mathsf{h}(j) \quad \text{for} \;\; 0 \le i < K$$



- Monotonically increasing

$$\mathsf{H}(K-1) = \sum_{j=0}^{K-1} \mathsf{h}(j) = M \cdot N$$

Last entry of
Cum. histogram

Total number of
pixels in image

Point Operation

- Point operations changes a pixel's intensity value.

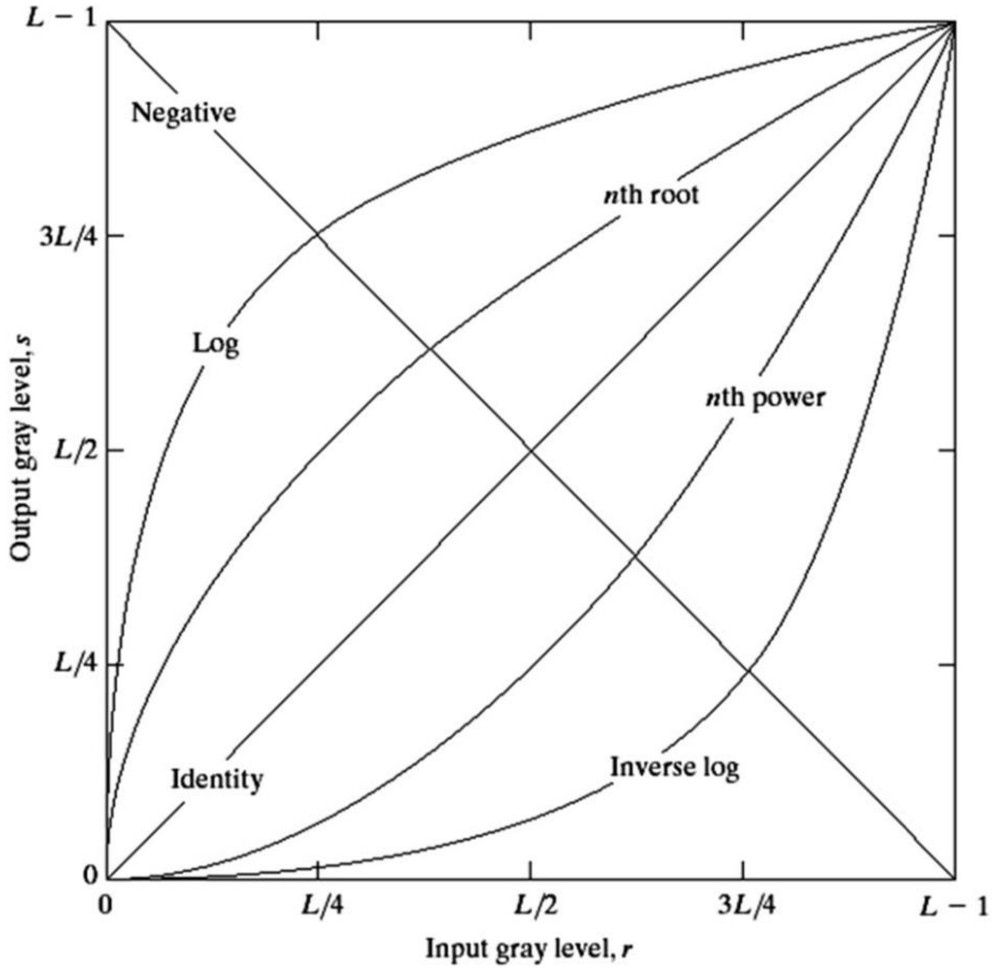
$$a' \leftarrow f(a)$$

$$I'(u,v) \leftarrow f\big(I(u,v)\big)$$

- New pixel intensity depends on
  - Pixel's previous intensity I(u, v)
  - Mapping function f( )
- Does not depend on
  - Pixel's location (u, v)
  - Intensities of neighboring pixels

Basic Grey Level Point Operation

- 3 most common gray level operation
  - Linear
    - Negative/Identity
  - Logarithmic
    - Log/Inverse log
  - Power law
    - nth power/nth root



Power Law Transformations

- Power law transformations have the form

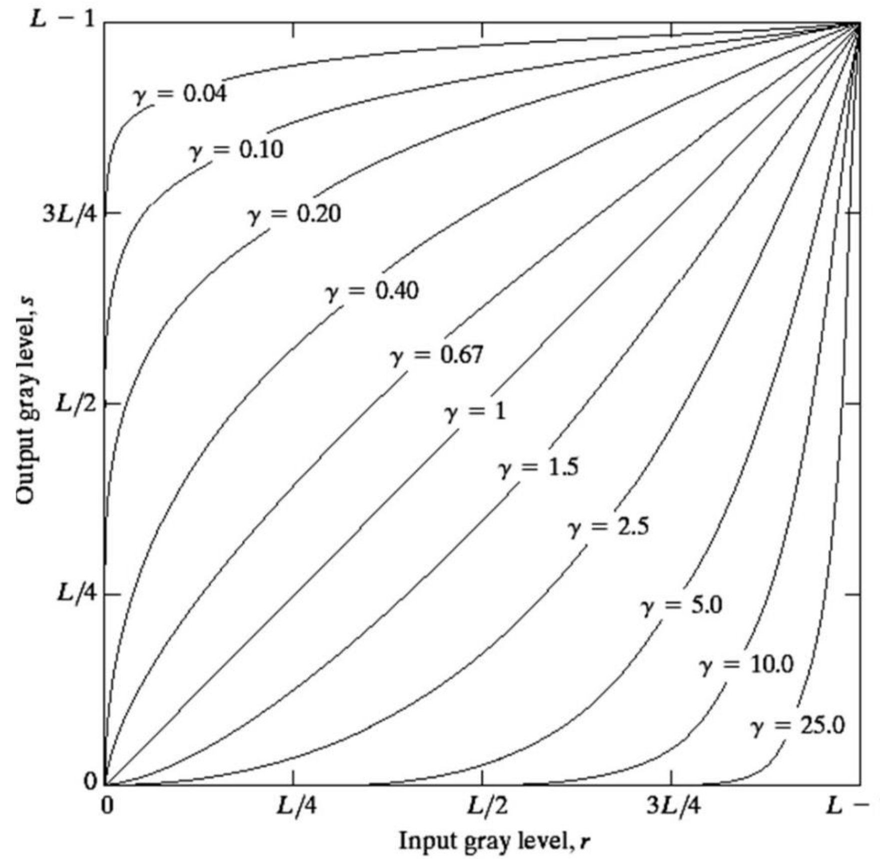
$$s = c * r^{\gamma}$$

Power

Old pixel value

New pixel value

Constant



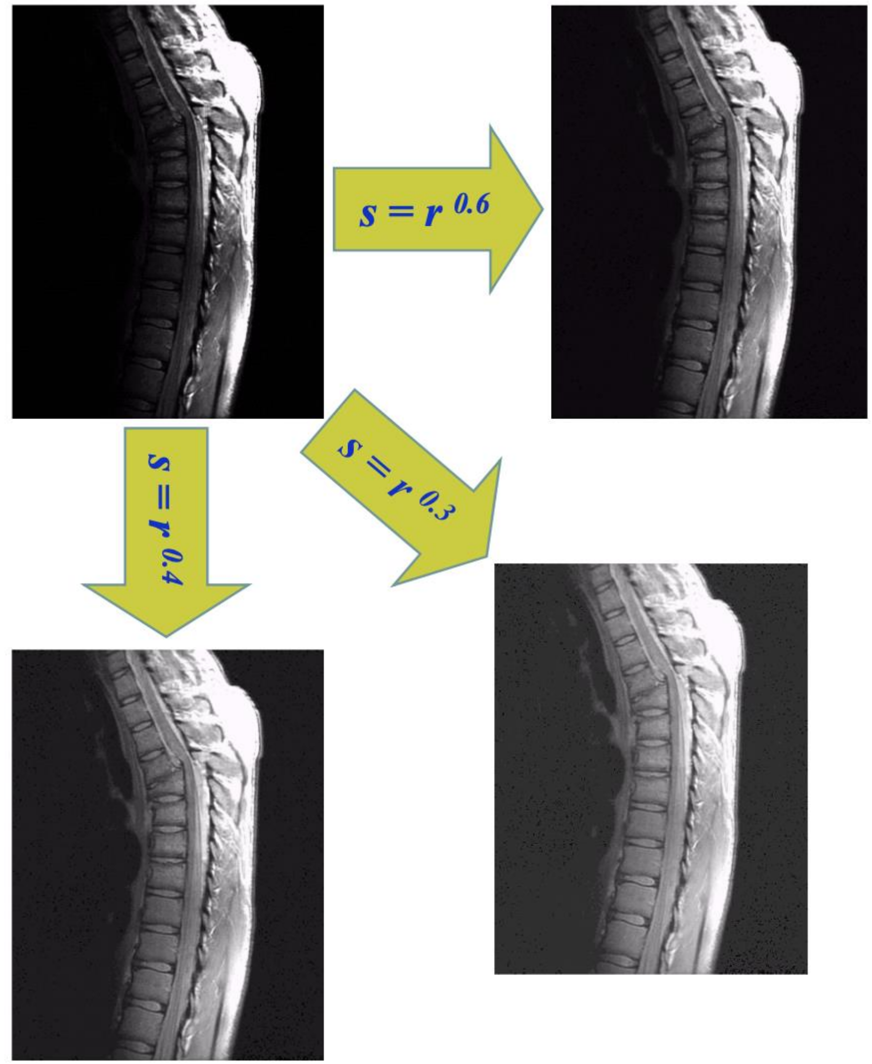
- Map narrow range of dark input values into wider range of output values or vice versa

- Varying γ gives a whole family of curves

Power Law Example

- Magnetic Resonance (MR) image of fractured human spine

- Different power values highlight different details

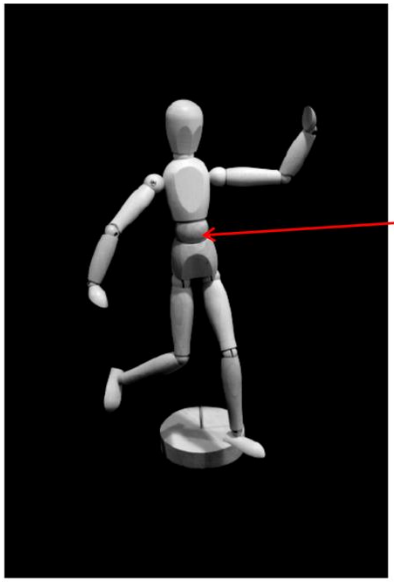


Intensity Windowing

- A clamp operation, then linearly stretching image intensities to fill possible range
- To window an image in [a,b] with max intensity M

$$f(p) = \begin{cases} 0 & \text{if } p < a \\ M \times \frac{p-a}{b-a} & \text{if } a \le p \le b \\ M & \text{if } p > b \end{cases}$$

Original Image



Contrasts easier to see

Windowed Image

Automatic Contrast Adjustment

- Modify pixel intensities such that the available range of range of pixels is full covered.
  - Algorithm
    - Find high and lowest pixel intensities, a_{low} and b_{high}
    - Linear stretching the intensity range.

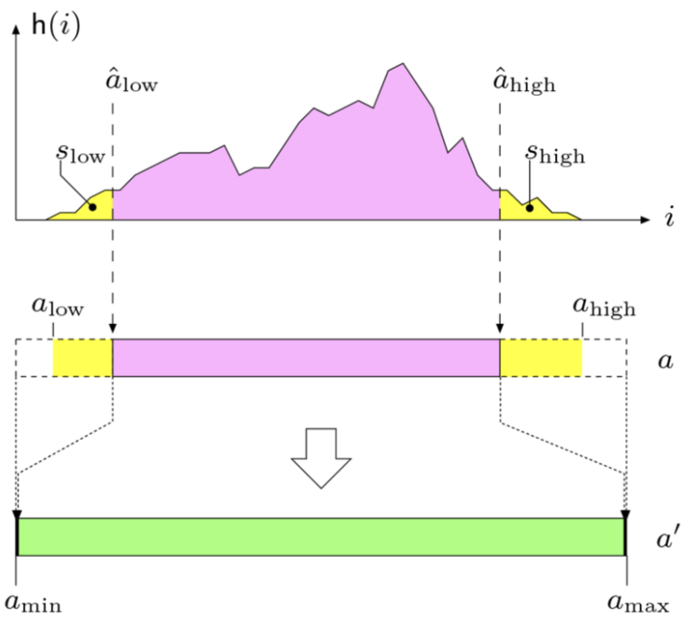
$$f_{\text{ac}}(a) = a_{\min} + (a - a_{\text{low}}) \cdot \frac{a_{\max} - a_{\min}}{a_{\text{high}} - a_{\text{low}}}$$

If a_{min} = 0 and a_{max} = 255

$$f_{\text{ac}}(a) = (a - a_{\text{low}}) \cdot \frac{255}{a_{\text{high}} - a_{\text{low}}}$$

Modified Contrast Adjustment

- Better to map only certain range of values.
- Get rid of tails (usually noises), with predefined percentiles s_{low} and s_{high}

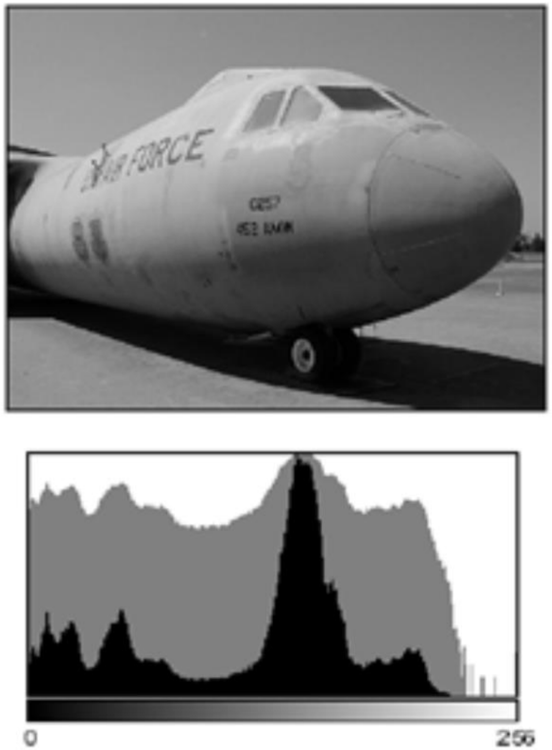


$$\hat{a}_{\text{low}} = \min\{\, i \mid \mathsf{H}(i) \geq M \cdot N \cdot s_{\text{low}} \}$$

$$\hat{a}_{\text{high}} = \max\{\, i \mid \mathsf{H}(i) \leq M \cdot N \cdot (1 - s_{\text{high}}) \}$$

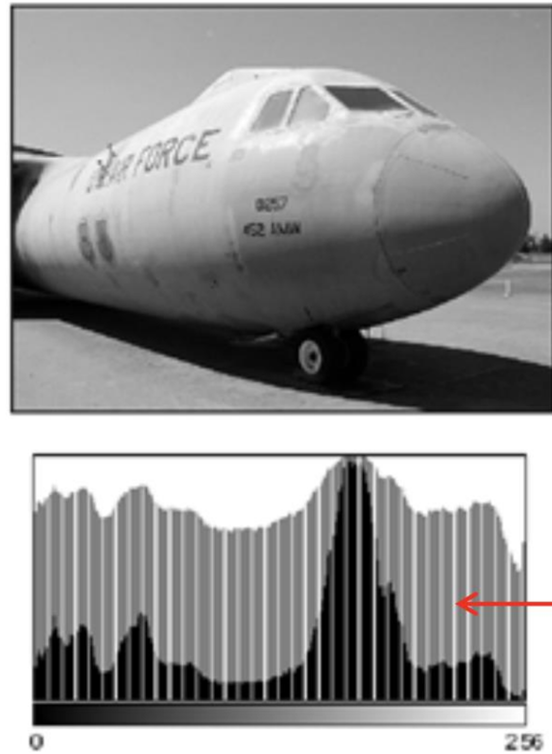
$$f_{\text{mac}}(a) = \begin{cases} a_{\text{min}} & \text{for } a \leq \hat{a}_{\text{low}} \\ a_{\text{min}} + (a - \hat{a}_{\text{low}}) \cdot \dfrac{a_{\text{max}} - a_{\text{min}}}{\hat{a}_{\text{high}} - \hat{a}_{\text{low}}} & \text{for } \hat{a}_{\text{low}} < a < \hat{a}_{\text{high}} \\ a_{\text{max}} & \text{for } a \geq \hat{a}_{\text{high}} \end{cases}$$

Effects of Automatic Contrast Adjustment



(a)

Original



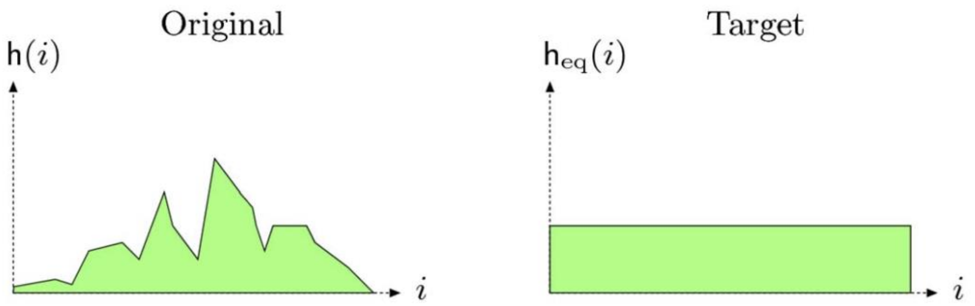
(b)

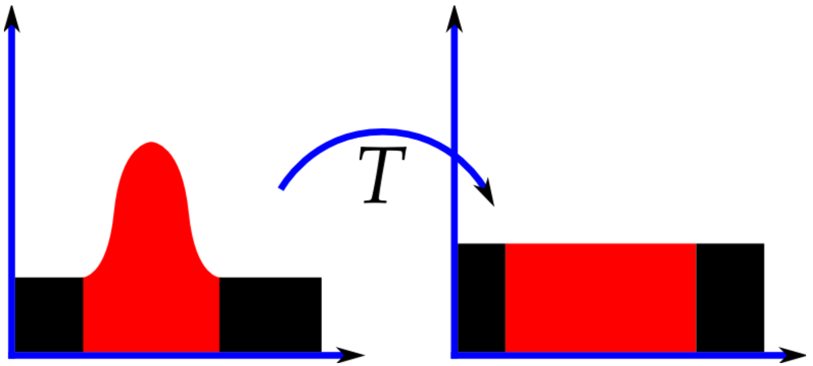
Result of automatic Contrast Adjustment

Linearly stretching range causes gaps in histogram

Histogram Equalization

- Adjust 2 different images to make their histograms (intensity distributions) similar
- Apply a point operation that changes histogram of modified image into uniform distribution.





Histogram Equalization: Algorithm

- Normalized Histogram

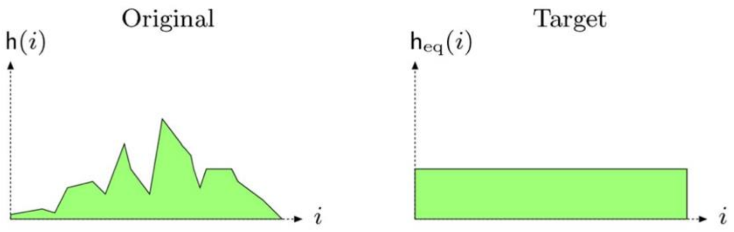
$$p_x(i) = p(x = i) = \frac{n_i}{n}, \quad 0 \le i < L$$

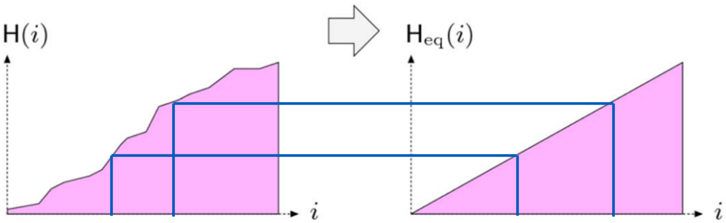
- Cumulated Normalized Histogram

$$\mathrm{cdf}_x(i) = \sum_{j=0}^{i} p_x(x = j),$$

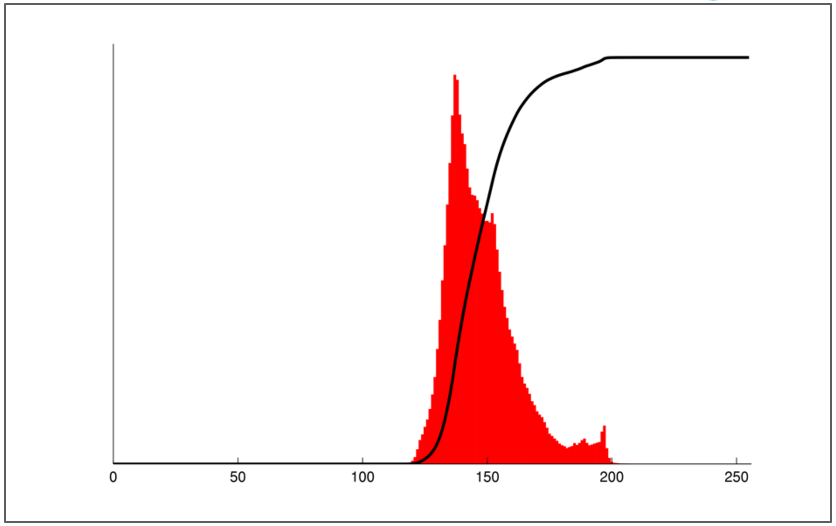
- A transform for histogram equalization, WHY?

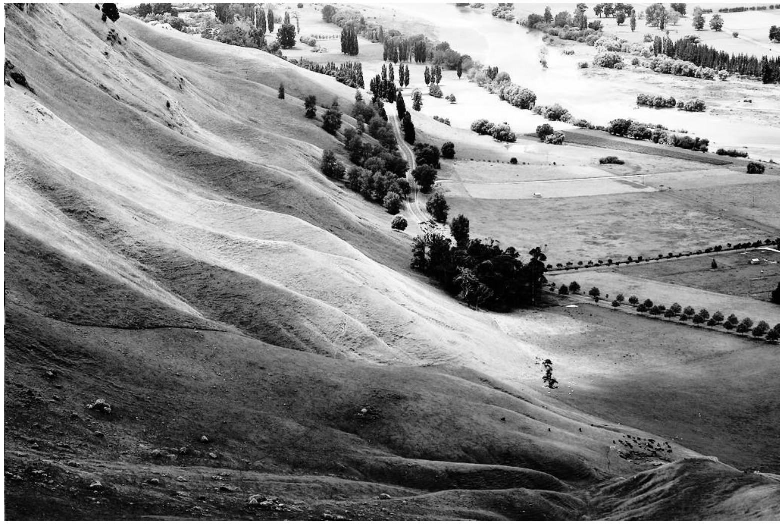
$$y = T(k) = \mathrm{cdf}_x(k)$$

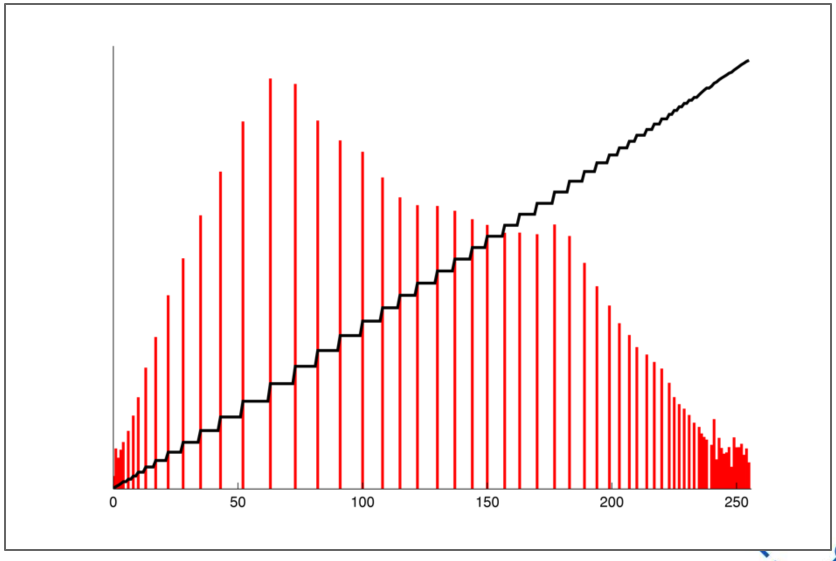




Histogram Equalization Example







Content

- Image Pyramids
  - Gaussian, Laplacian
  - Convolution and Transposed Convolution
- Image Histogram
  - Equalization, Matching
  - Image Enhancement
  - Histogram Equalization