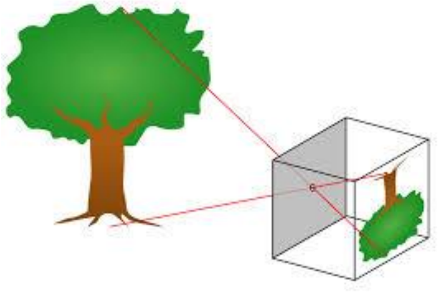
Image Digitization: 3 Stages

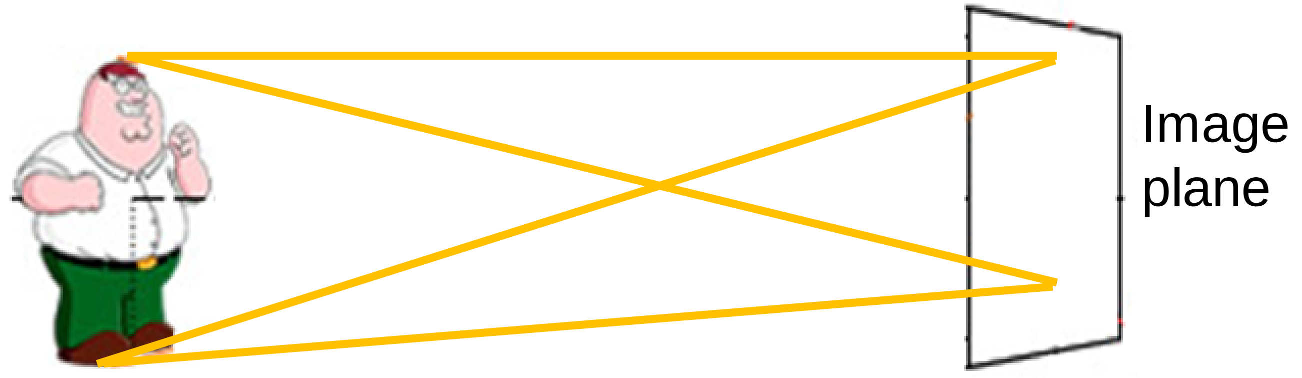
1) Transforming the **3D** world into **2D** image
   – Perspective Projection (Optics, Continuous)



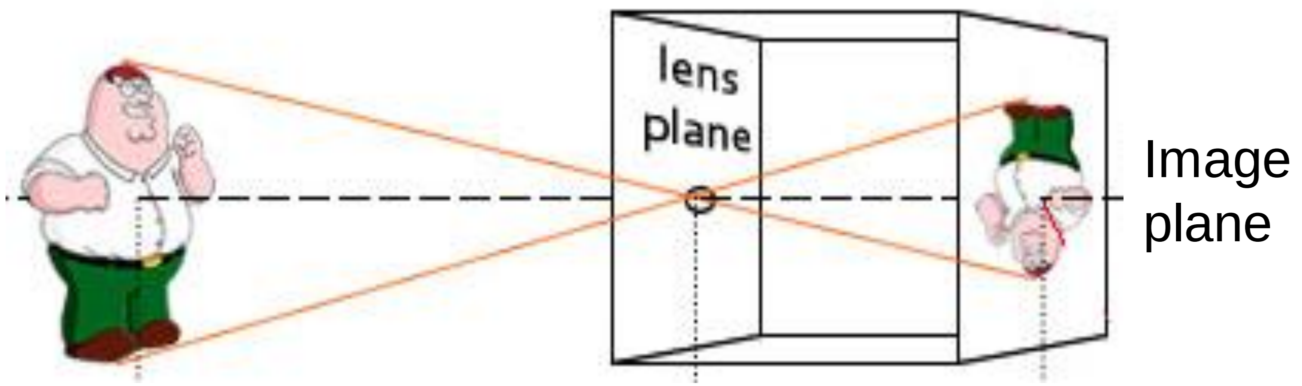
2) Sampling the Image Plane
   – Finite number of **Pixels**

3) Quantizing the color/gray-level
   – Finite number of colors (e.g. 8 bits per color)

Pinhole Camera (Camera Obscura (Latin) = Dark Room)

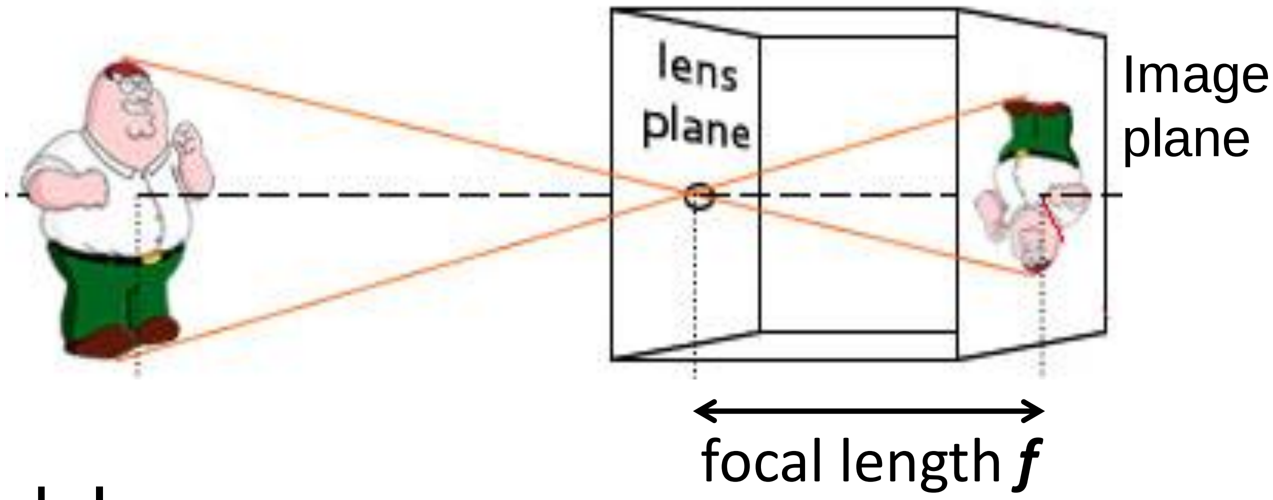


- No Image is generated when we place an image plane in the world



- To create an image, each image location should get a ray from a single point. This is done by blocking all rays except one…

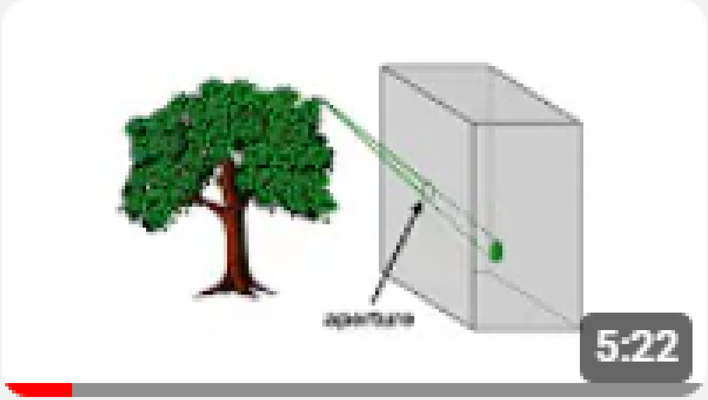
Pinhole Camera (Camera Obscura (Latin) = Dark Room)



- Pinhole model:
  - Captures pencil of rays – all rays through a single point
  - The point (pinhole) is called Center of Projection (COP)
  - The image is formed on the Image Plane
  - Focal length *f* is distance from COP to Image Plane

5 Minute Video Clips

- 12) Perspective projection
- 13) Perspective projection: Part 2 – the math!

12 

Perspective projection in 5 minutes

Graphics in 5 Minutes • 24K views • 2 years ago

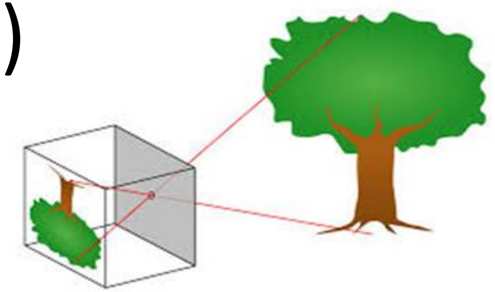
13 

Perspective projection in 5 minutes: Part 2 -- the math!

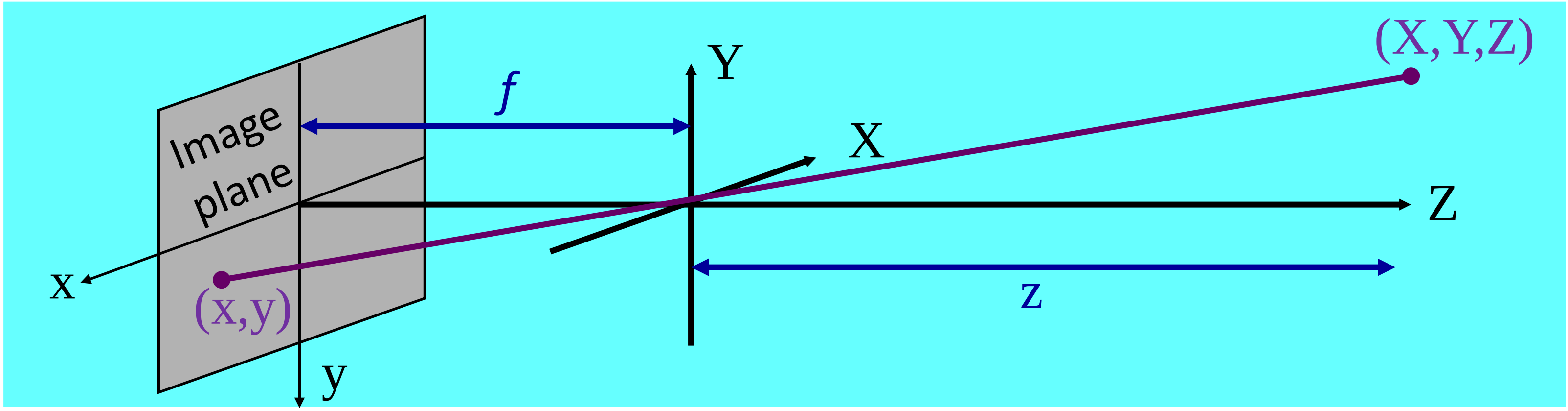
Graphics in 5 Minutes • 8.3K views • 2 years ago

Perspective Projection

- Transforming the 3D world (X, Y, Z) into 2D image (x, y)
  - Continuous Perspective Projection (optics)
  - All rays pass through one point (f = focal length)



Simple case: Aligned World axis (X, Y, Z) and Image axis (x, y)

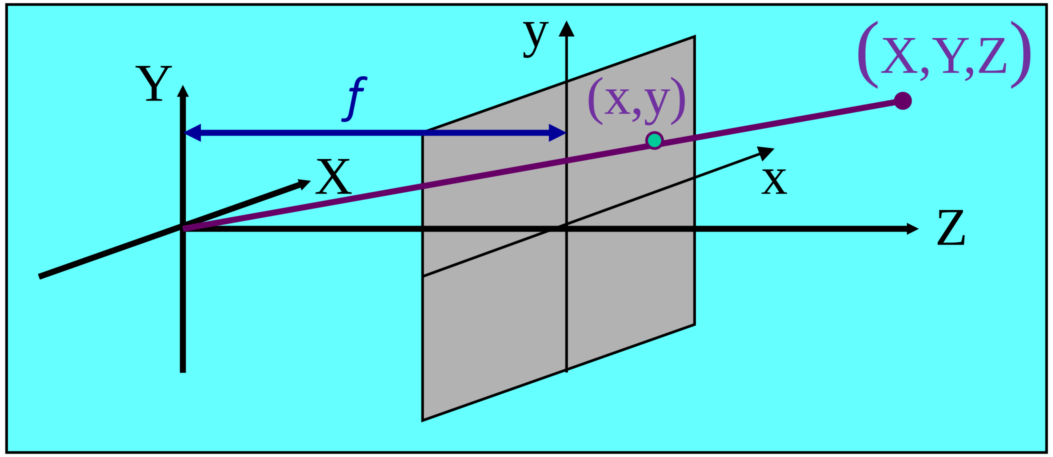


$$\frac{Y}{Z} = \frac{y}{f}$$

Similar Triangles

$$x = \frac{f}{Z} X$$

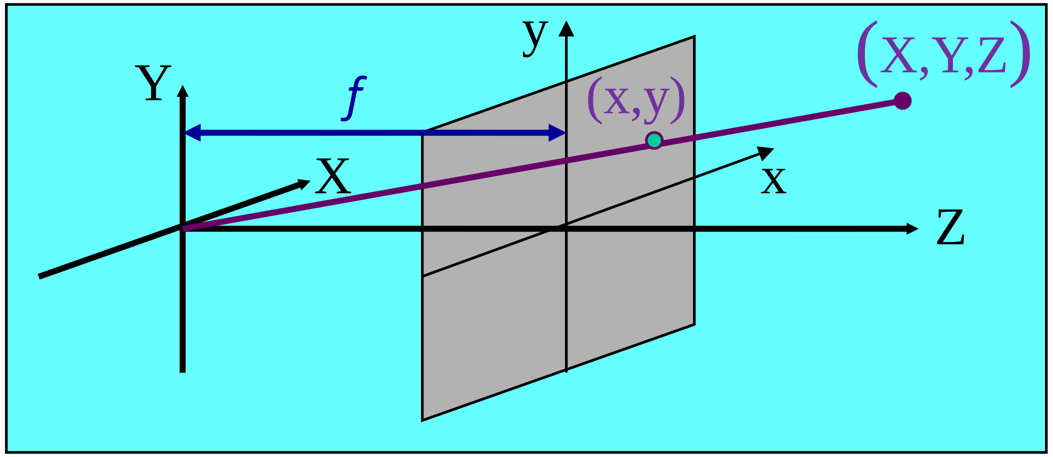
$$y = \frac{f}{Z} Y$$



Perspective Projection

- World to Camera Transformation

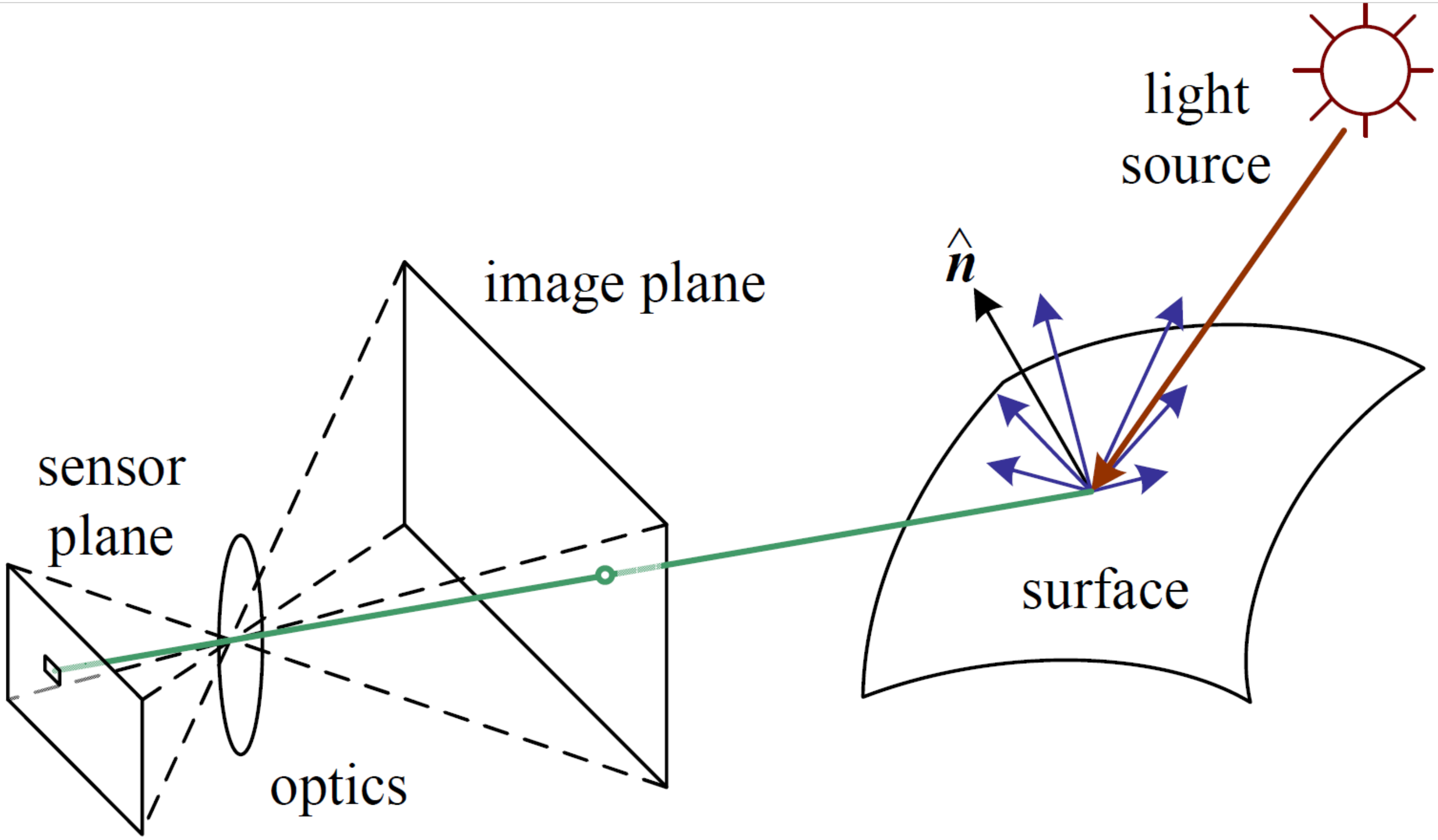
$$x = \frac{f}{Z} X \qquad\qquad y = \frac{f}{Z} Y$$



- Only when world axis (X, Y, Z) and Image axis (x, y) are aligned:
  - X is parallel to x
  - Y is parallel to y
  - Same units

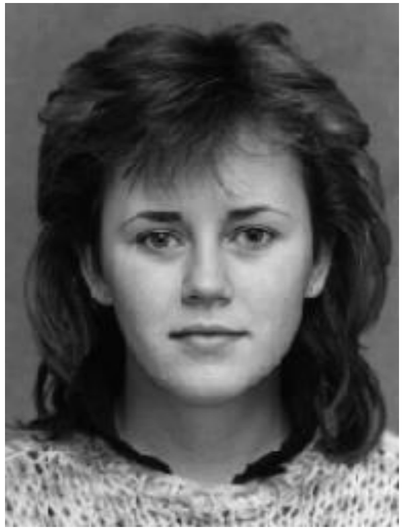
- In the general case, there is a transformation matrix between world axis and camera axis.

Summary: First Stages of Image Acquisition
Analog Light



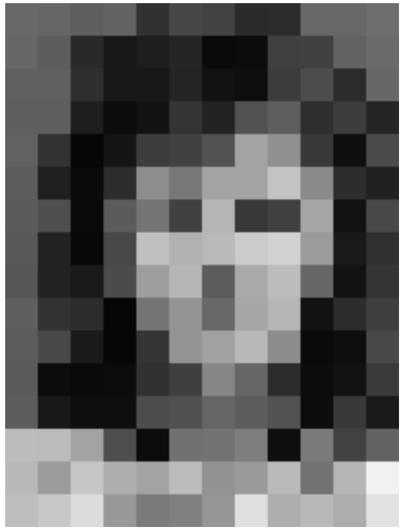
Spatial Sampling to Pixels

- Sampling the Image Plane
  - Finite number of **Pixels**
  - Do we always want maximum number of pixels?

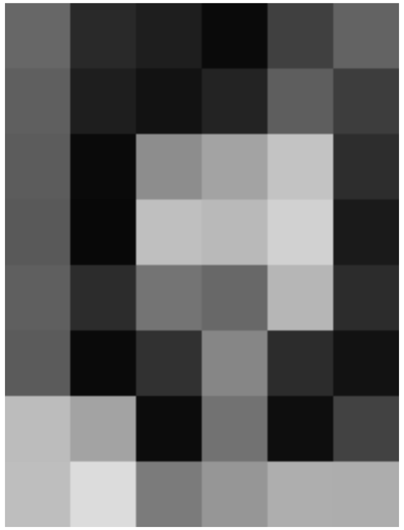


256 lines

64 lines



16 lines

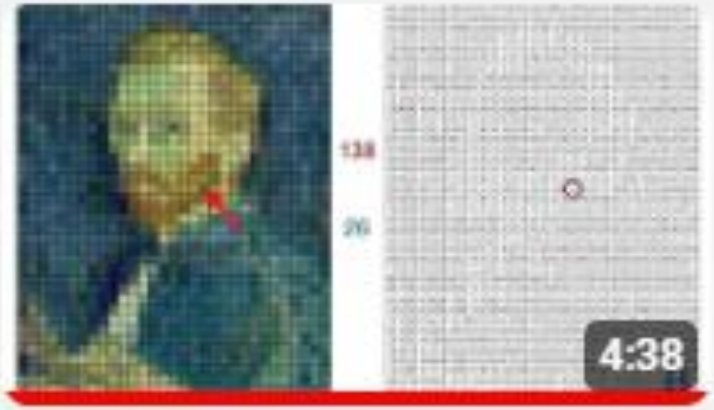


8 lines

5 Minute Video Clips

- 4) Images in 5 minutes

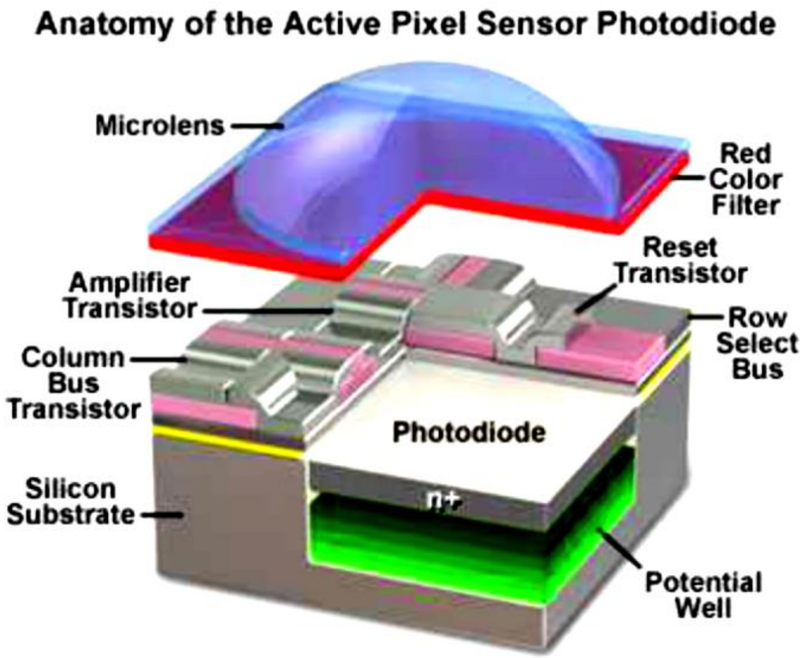
4

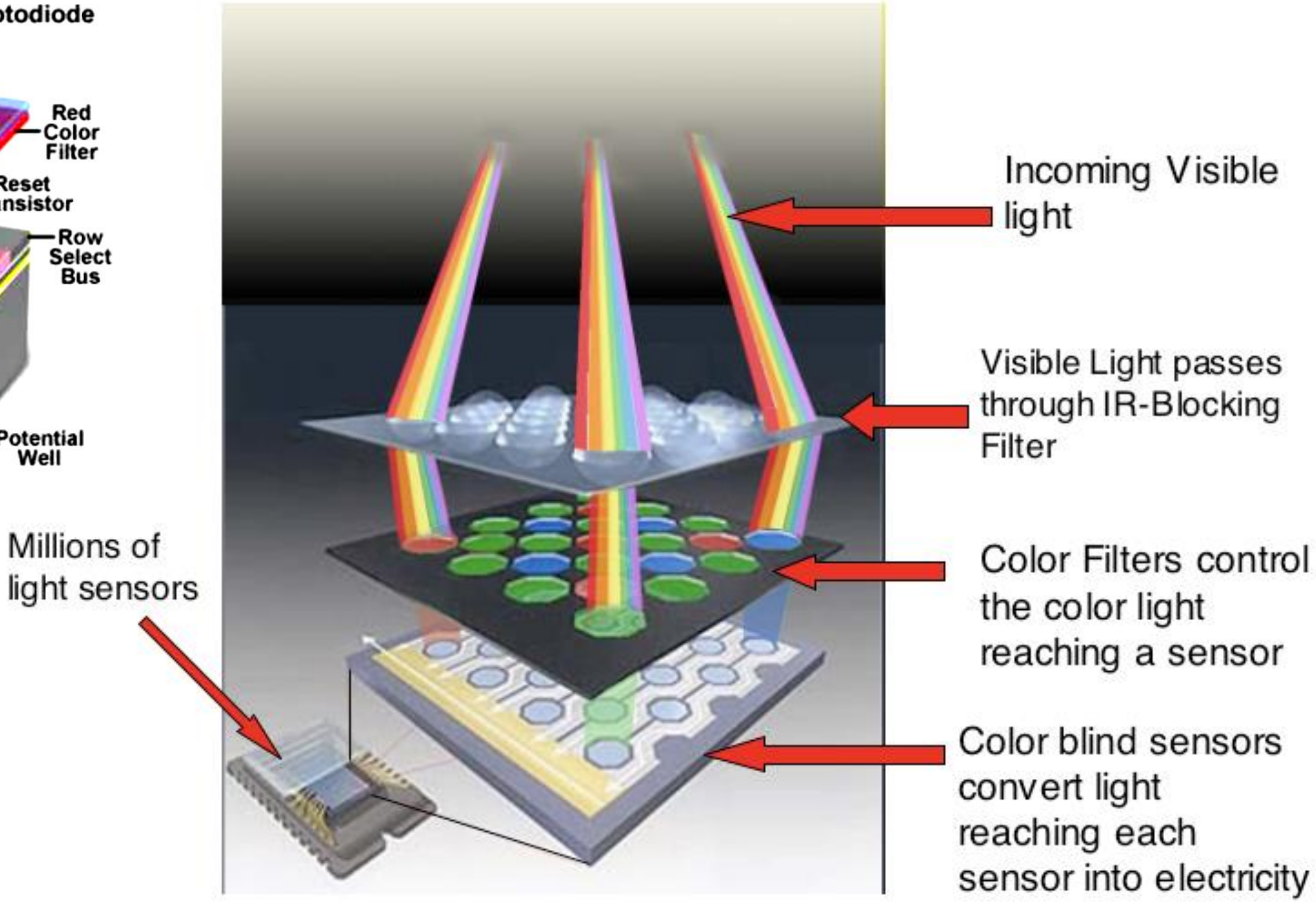


Images in 5 minutes: The Case of the Splotched Van Gogh, Part 1

Graphics in 5 Minutes • 4.4K views • 2 years ago

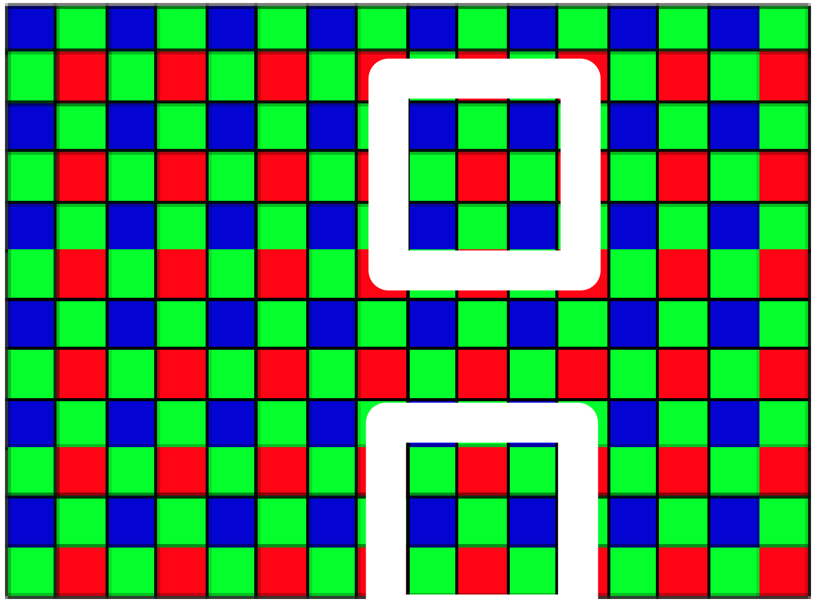
RGB Inside the Camera





Bayer Filter

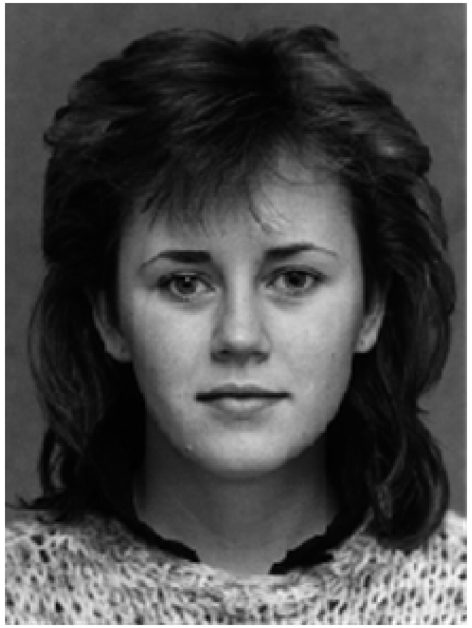
- In 1975, Bruce Bayer invents the color filter array, used in most digital camera.
- ¼ pixels detect red; ¼ pixels blue; ½ pixels green;
- The camera <u>invents</u> 2 missing colors in pixels. How?



- Demosaicing: Invents missing colors
- Many methods, mostly proprietary
- A possible (bad) method:
  - Average 2 or 4 neighbors

Color/Gray-level Quantization

- Quantizing the color/gray-level
  - Finite number of colors



256 Levels

8 Levels

4 Levels

2 Levels

Digital Pictures

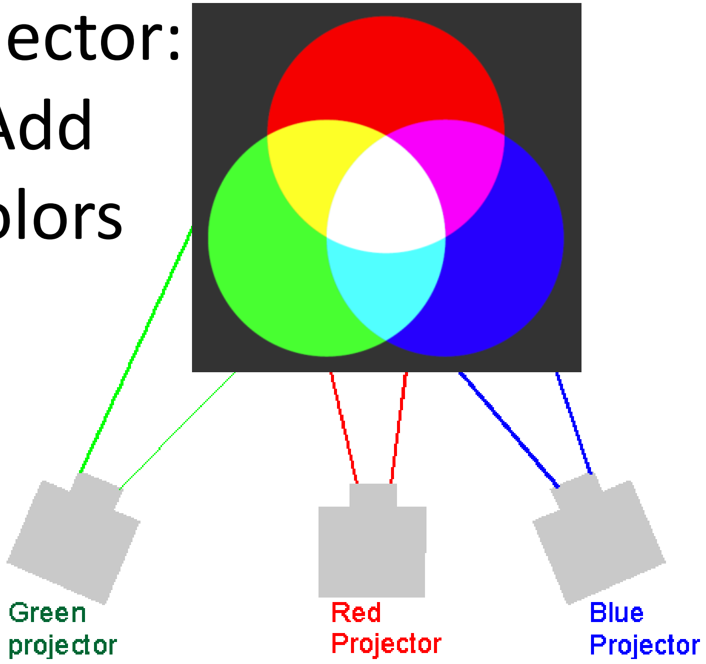
- A Matrix of numbers (Greylevel image)
- A Matrix of triplets (RGB Color, etc.)

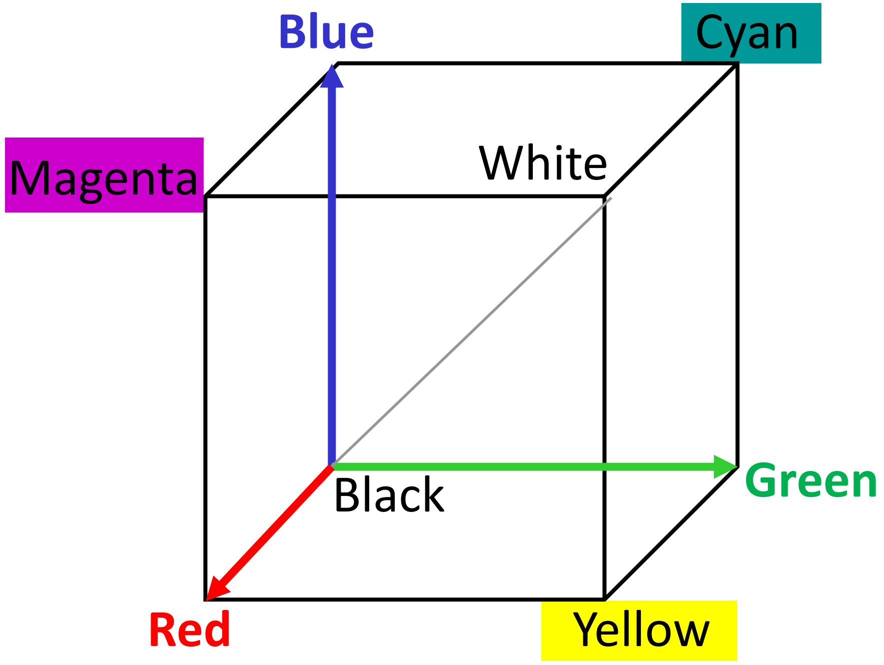
2	4	5	6	7	8	9	10	11	12	11	10	9	8	7
3	5	6	7	8	9	10	11	12	13	12	11	10	9	8
4	6	7	8	9	10	11	12	13	14	13	12	11	10	9
5	7	8	9	10	11	12	13	14	15	14	13	12	11	10
6	8	9	10	11	12	13	14	15	16	15	14	13	12	11
7	9	10	11	12	13	14	15	16	17	16	15	14	13	12
8	10	11	12	13	14	15	16	17	18	17	16	15	14	13
9	11	12	13	14	15	16	17	18	19	18	17	16	15	14
10	12	13	14	15	16	17	18	19	20	19	18	17	16	15
9	11	12	13	14	15	16	17	18	19	18	17	16	15	14
8	10	11	12	13	14	15	16	17	18	17	16	15	14	13
7	9	10	11	12	13	14	15	16	17	16	15	14	13	12
6	8	9	10	11	12	13	14	15	16	15	14	13	12	11
5	7	8	9	10	11	12	13	14	15	14	13	12	11	10
4	6	7	8	9	10	11	12	13	14	13	12	11	10	9
3	5	6	7	8	9	10	11	12	13	12	11	10	9	8
2	4	5	6	7	8	9	10	11	12	11	10	9	8	7
1	3	4	5	6	7	8	9	10	11	10	9	8	7	6

Color Spaces

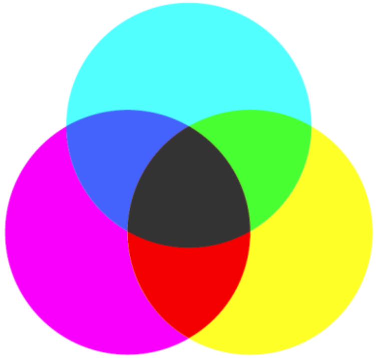
RGB (Camera, Projector - Add), CMYK (Print -Subtract), YIQ (TV)

Projector:
Add
colors





Ink:
Absorb
colors



For Color to B/W TV

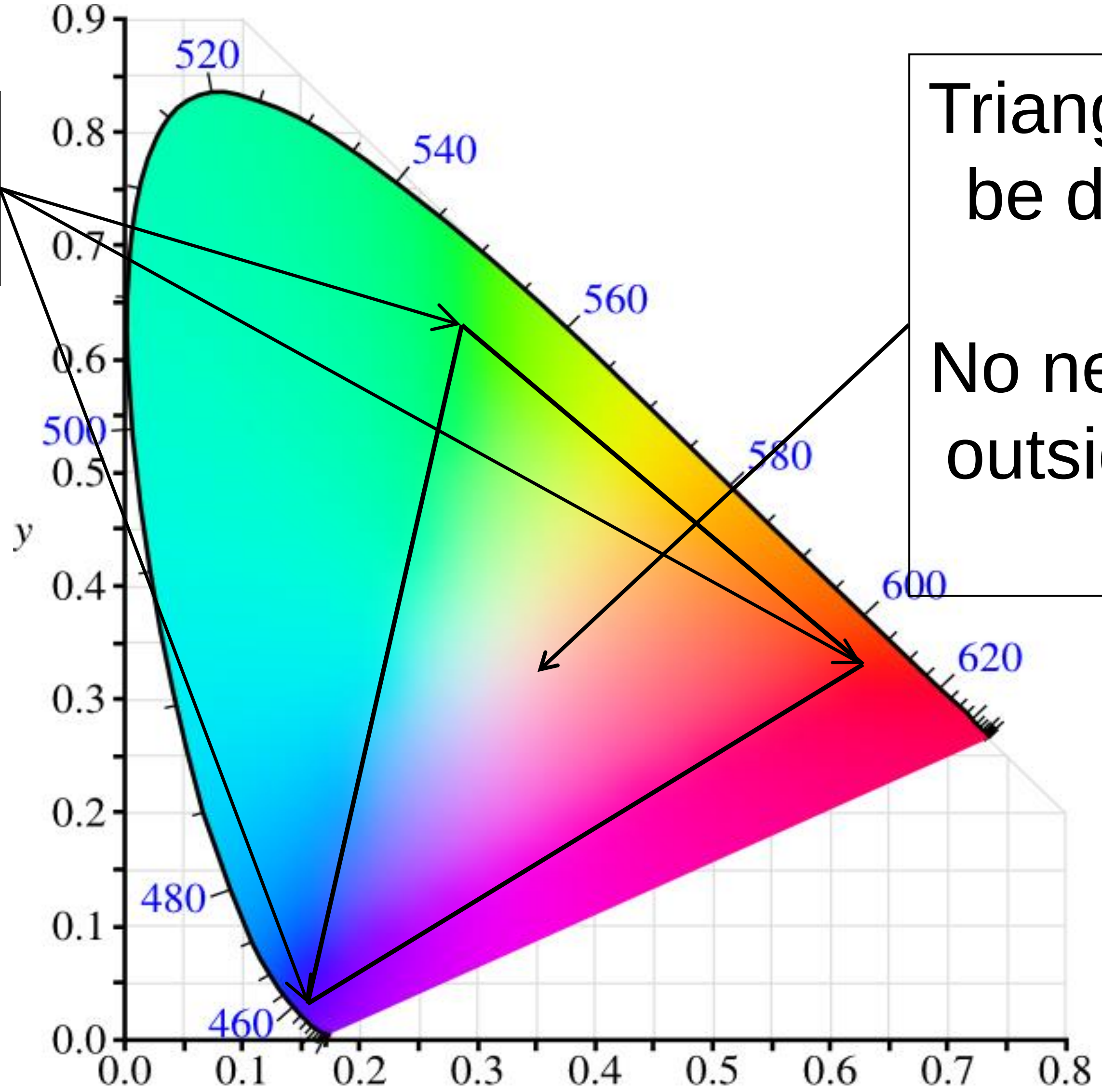
Y - Luminance

$$\begin{vmatrix} Y \\ I \\ Q \end{vmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

CIE Chromaticity Diagram (1931)

Boundary: Spectral Colors (Single Wavelength)

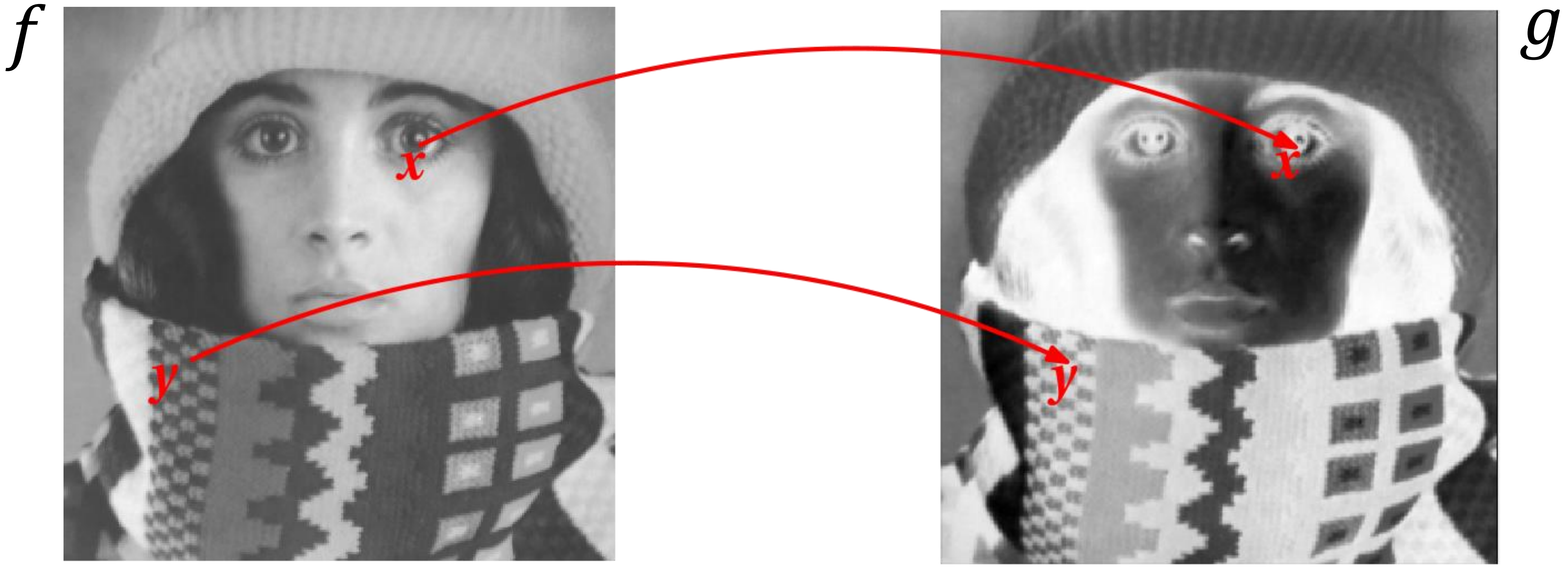
Colors of LED TV



Triangle: Colors that can be displayed (Gamut)

No new colors displayed outside Gamut (Display Limitation)

Point Operations



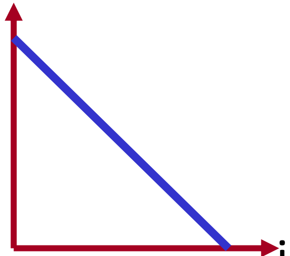
- New pixel value $g(x,y)$ based on the input pixel value $f(x,y)$
- E.g. $g(x,y)=255-f(x,y)$ (Negative)
- Operation depends only on pixel value (No location…)

Point Operations

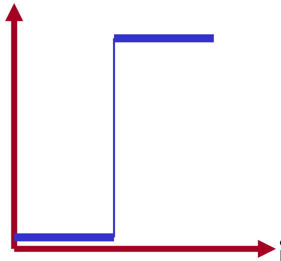
f

g

- In general, $g(x, y) = T(f(x, y))$
- $T(u) = 255 - u$ Negative



- $T(u) = \begin{cases} 0 & if\ u < 127 \\ 1 & if\ u > 127 \end{cases}$ Threshold

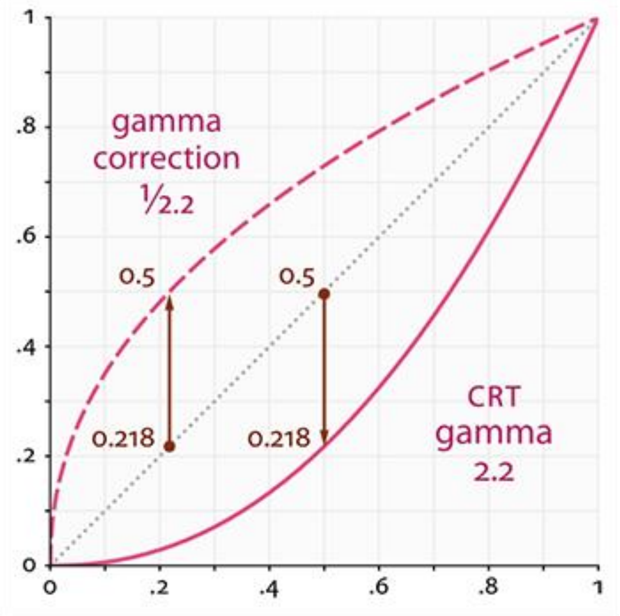


Point Operations - γ Correction

Gamma Correction is used to overcome non-linear responses of camera, display, and eyes

$$T(u) = Max \cdot \left(\frac{u}{Max}\right)^{\gamma}$$

When Max=1: $T(u) = u^{\gamma}$

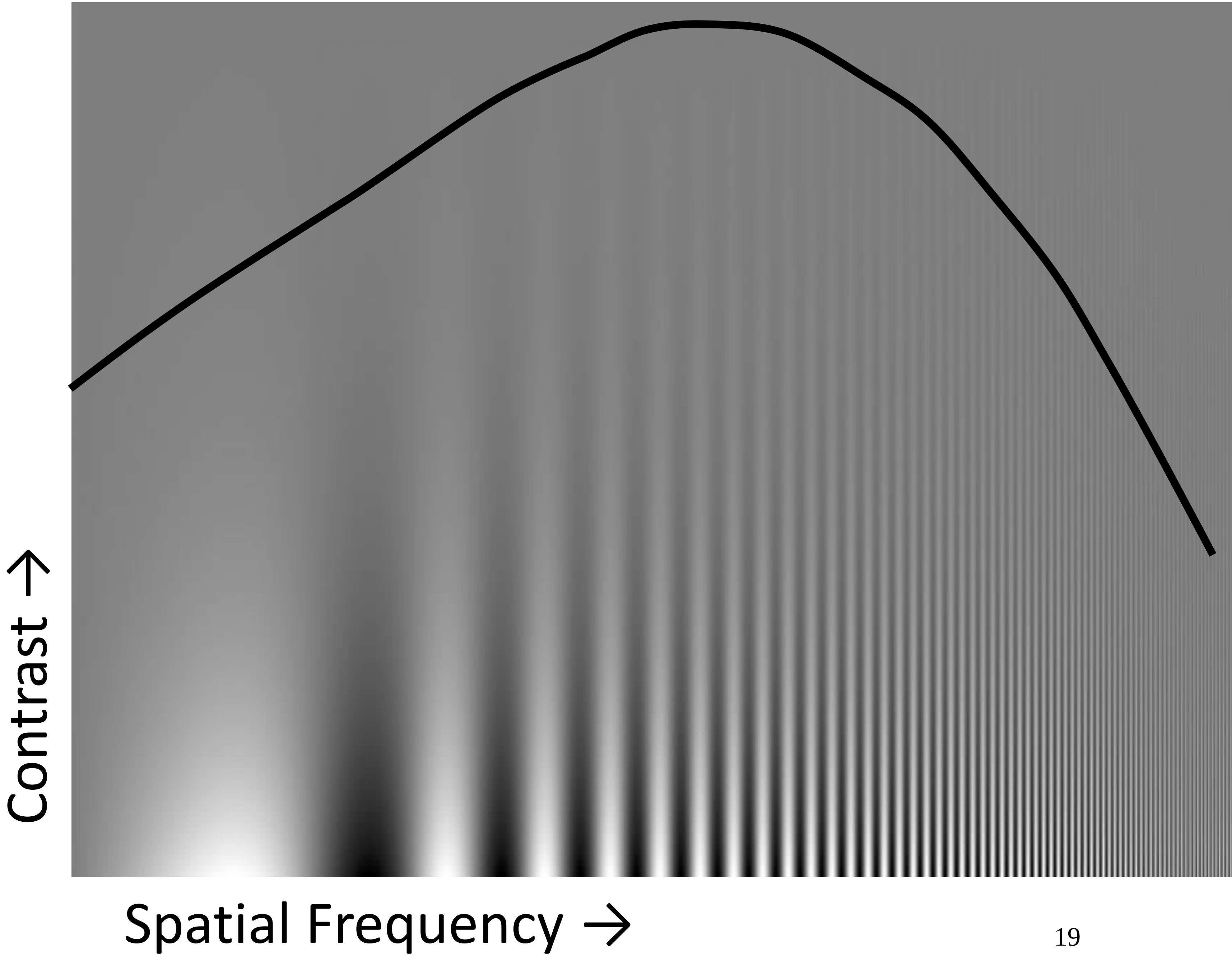


$\gamma = 1$

$\gamma = 1/2.2$

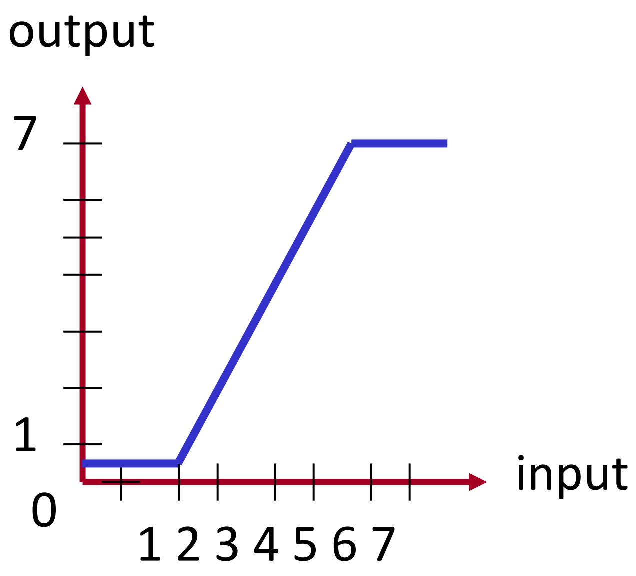
Eye Sensitivity

- Eye sensitivity as a function of frequency



Point Transformation - Look Up Table (LUT)

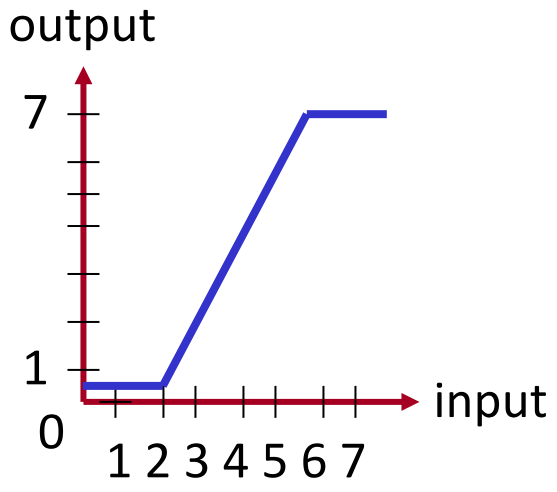
- LUT efficiently Represent transformations L from N to N (or to other). E.g.
  - L(0)=0
  - L(1)=0
  - L(2)=0
  - L(3)=2
  - L(4)=4
  - L(5)=6
  - L(6)=7
  - L(7)=7



Point Operation with LUT

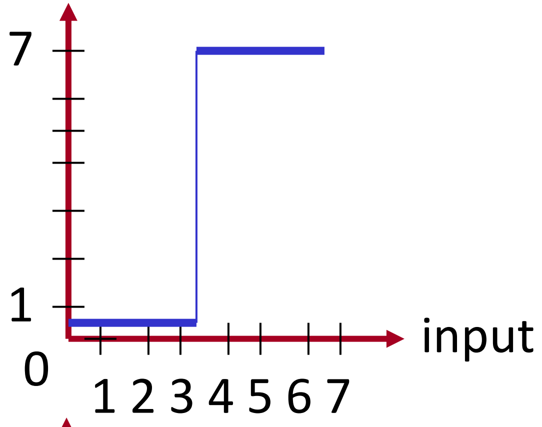
- Stretch

In	0	1	2	3	4	5	6	7
Out	0	0	0	2	4	6	7	7



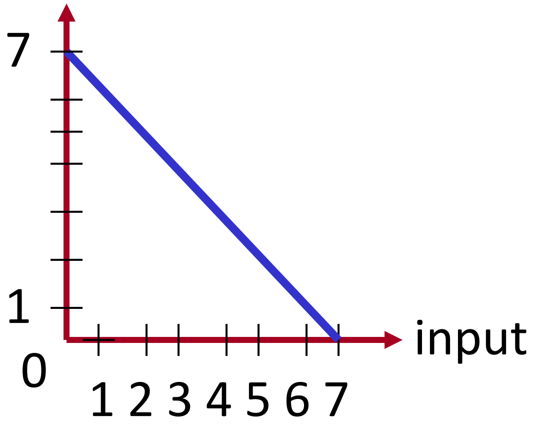
- Threshold

In	0	1	2	3	4	5	6	7
Out	0	0	0	0	7	7	7	7



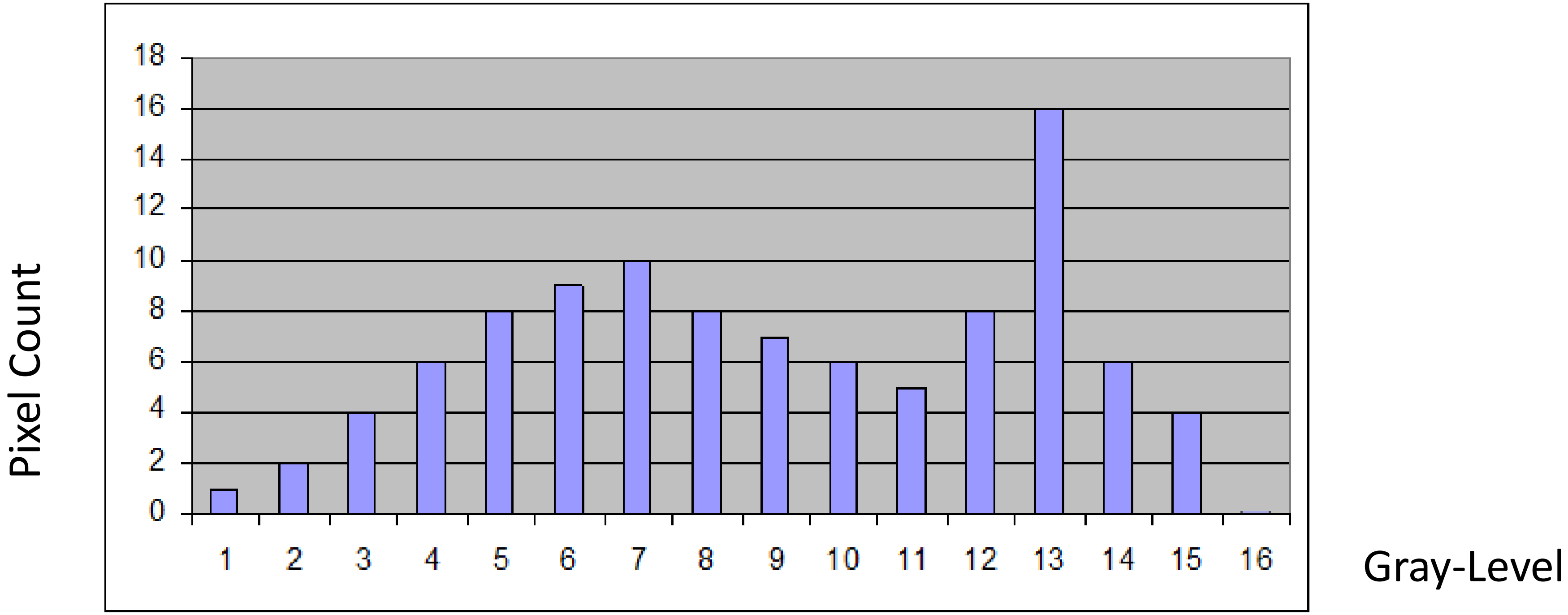
- Negative

In	0	1	2	3	4	5	6	7
Out	7	6	5	4	3	2	1	0



The Histogram

- Frequency counting of gray levels or colors
- Analogous to PDF – Probability Density Function

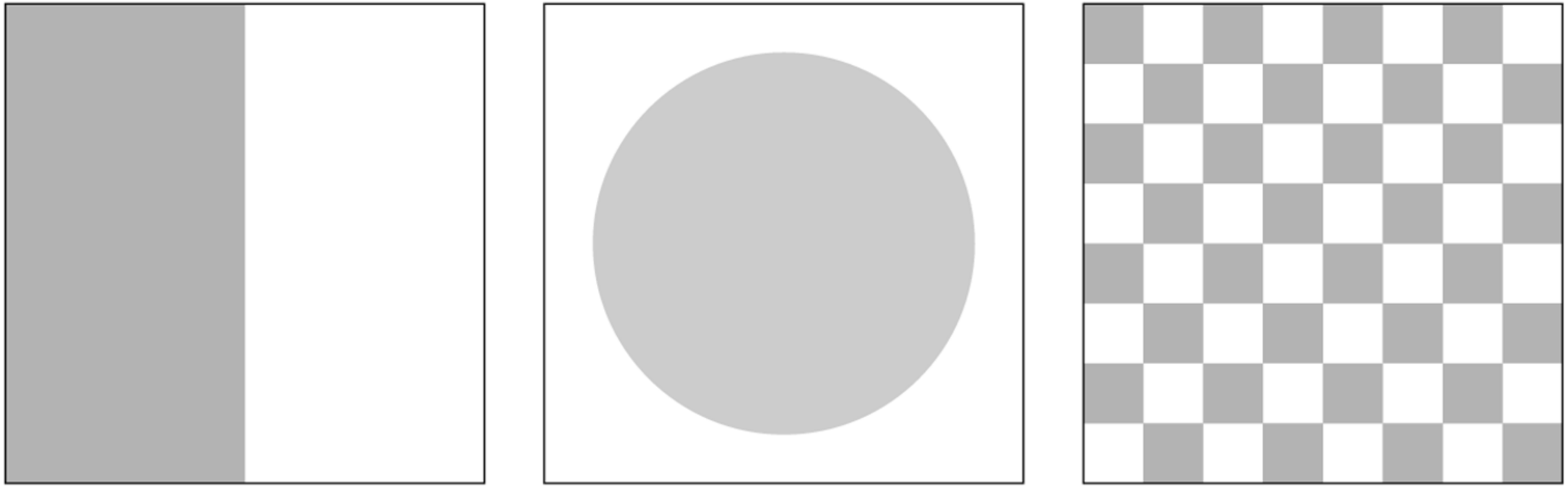


Pixel Count	1	2	4	6	8	9	**10**	8	7	6	5	8	**16**	6	4	0
Grey Level	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

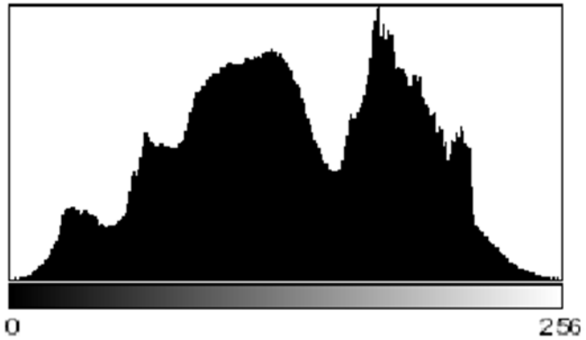
Common Histogram has 256 Grey Levels

Histogram is Invariant to Pixel Location

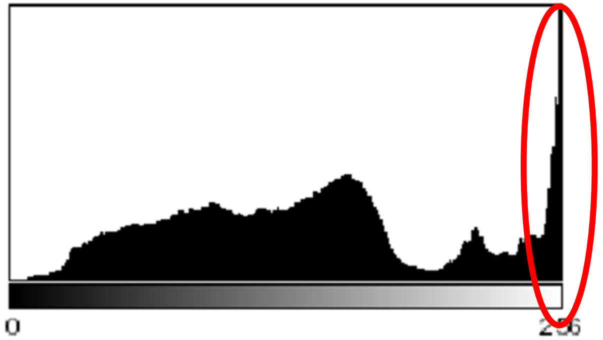
- Different pictures can give same histograms
- All three pictures below have same histogram. What is it?



Histogram & Exposure

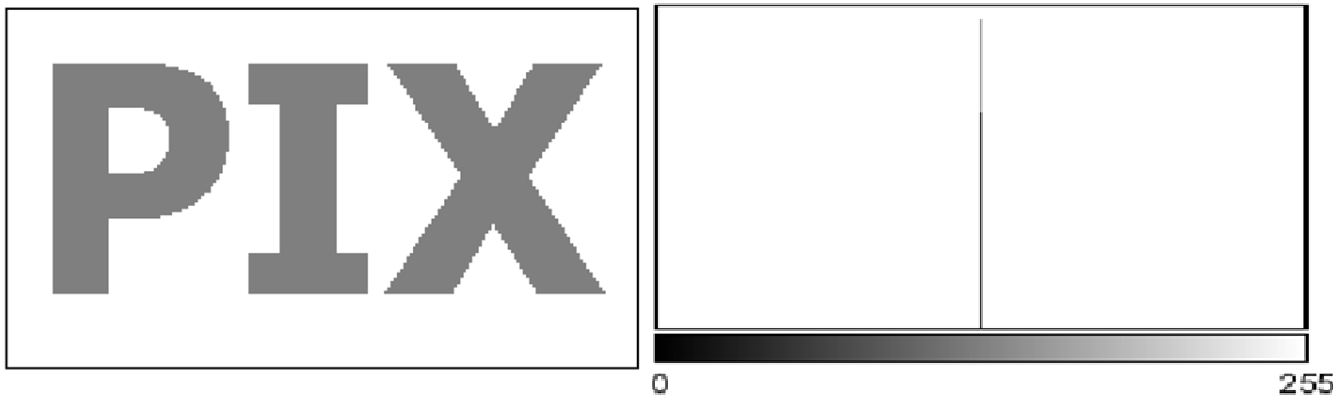


OK

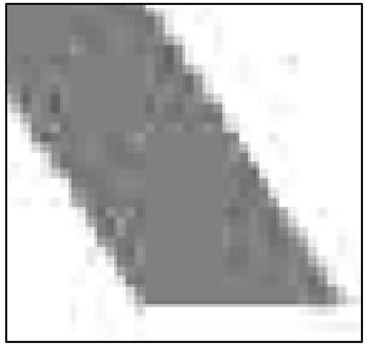


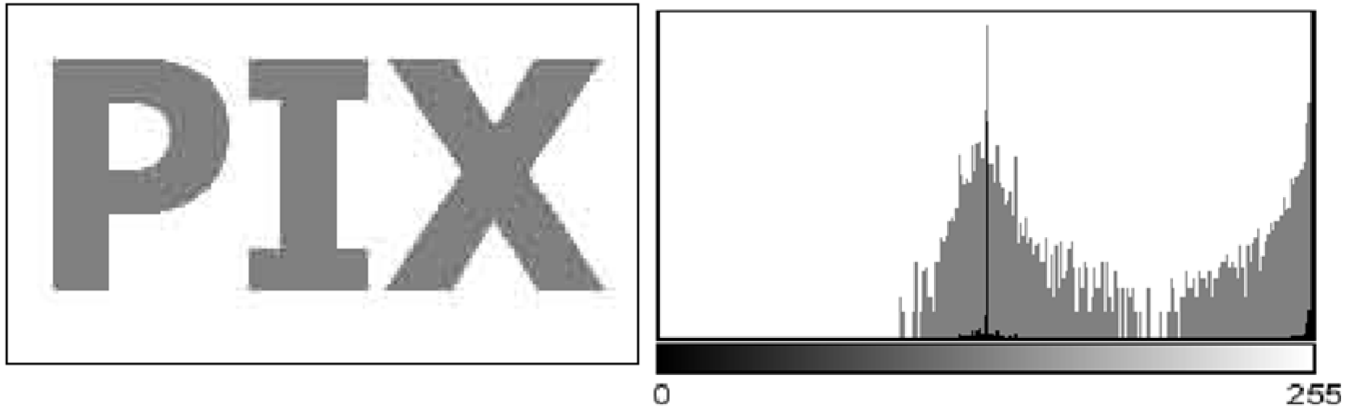
Over Exposed

JPG Compression Effects



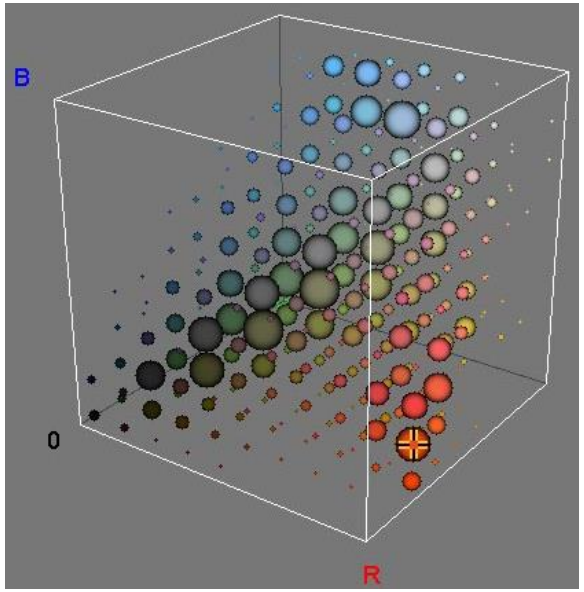
- Original Image





- Image after JPG compression

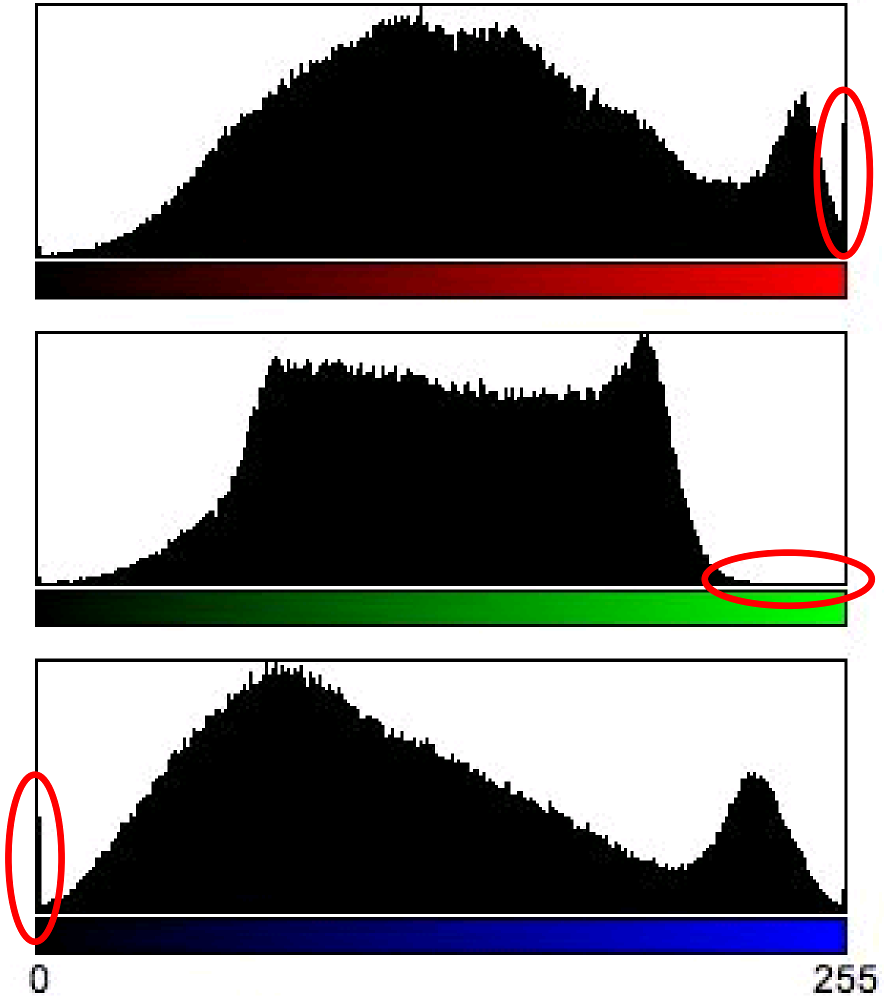
Color Histogram Options



3D Histogram
RGB Space

3D array
256^3~16.8M

22K times more than
3 histograms



Red Channel

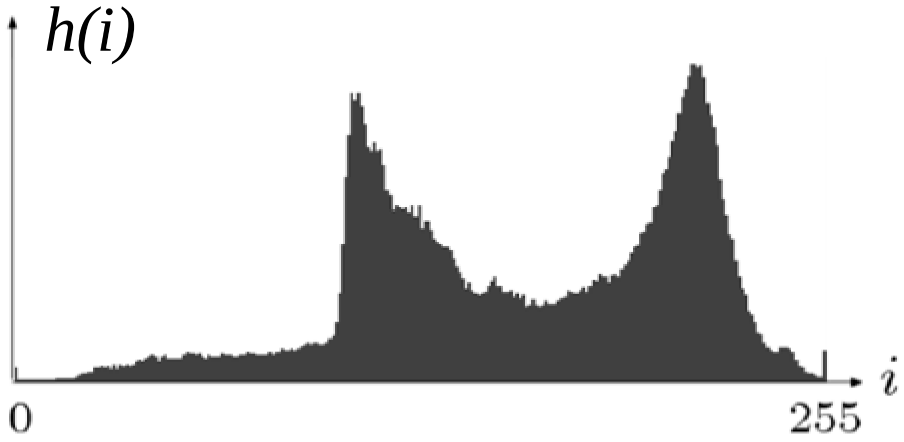
Green Channel

Blue Channel

Histogram: 1D array of 256 integers
3 colors (RGB): 768 integers

Cumulative Histogram

- Let $h(i)$ he a histogram. $S(i) = \sum_{j=0}^{i} h(j)$
- $S(0) = h(0);$ $S(255) =$ # of pixels in image
- Analogous to CDF – Cumulative Density Function

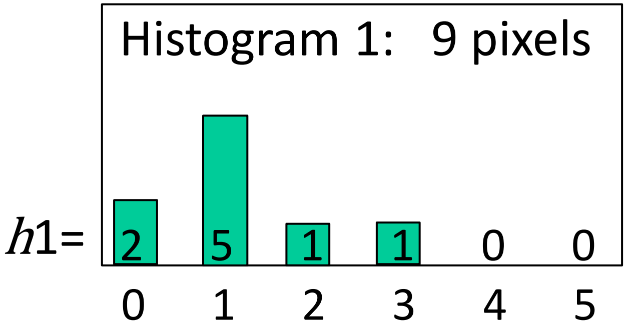


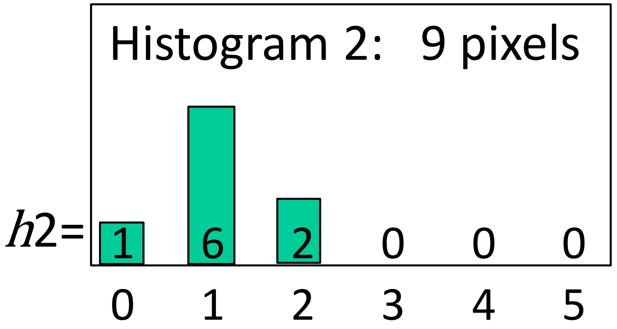
- What is the area under the curve of h ?
- What is the area under the curve of S ?

Histogram Application: Video Scene Cut Detection

- Similar images inside shots having similar histograms
  - Video Shot Cut Detection: same shot → similar histograms
- Compute distance between color histogram of successive frames

Distance Between Histograms (Distributions)



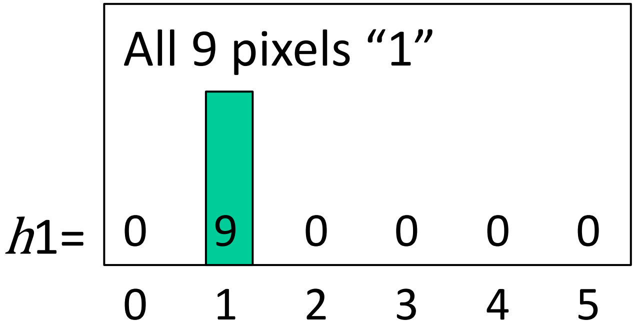


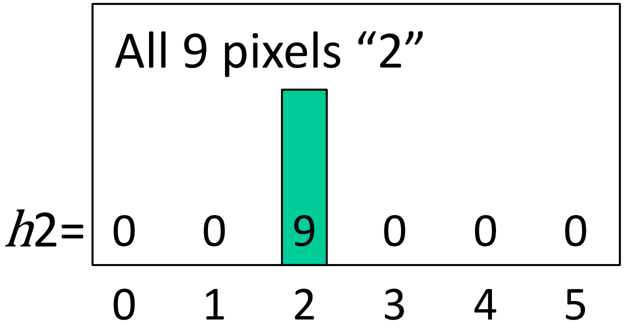
Vector Distance: $|h1 - h2| =$

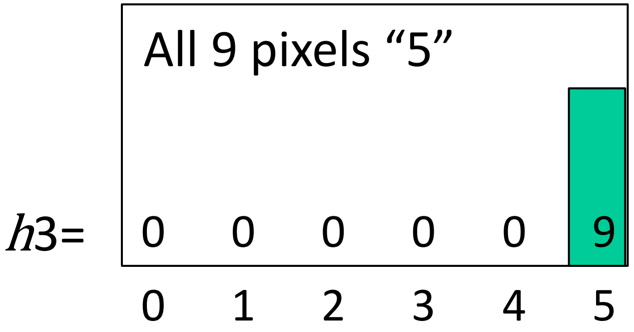
$$= |2-1| + |5-6| + |1-2| + |1-0|$$

$$= 4$$

But in some cases, this simple vector distance does not work:

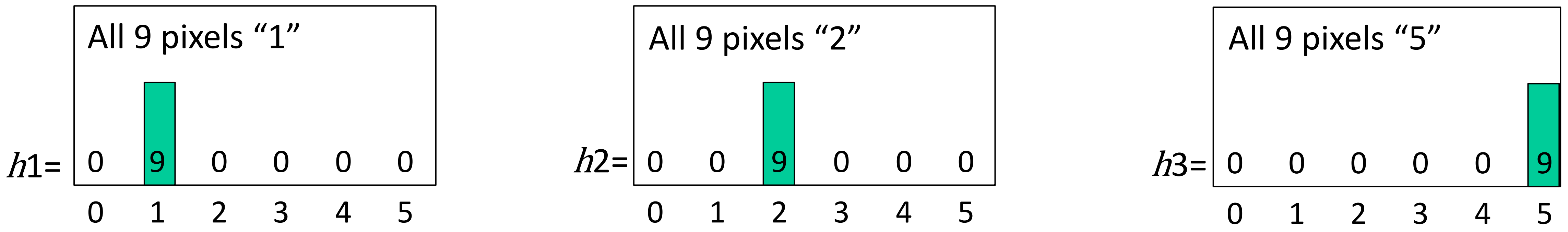




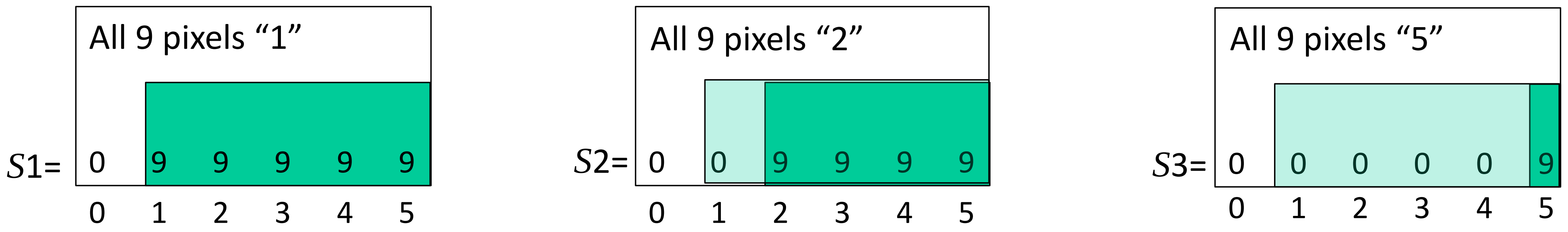


- All three have equal vector distance $|h1 - h2| = |h1 - h3| = 9 + 9 = 18$
- Seems wrong, since “2” is closer to “1” than “5”

Distance Between Histograms (Distributions)



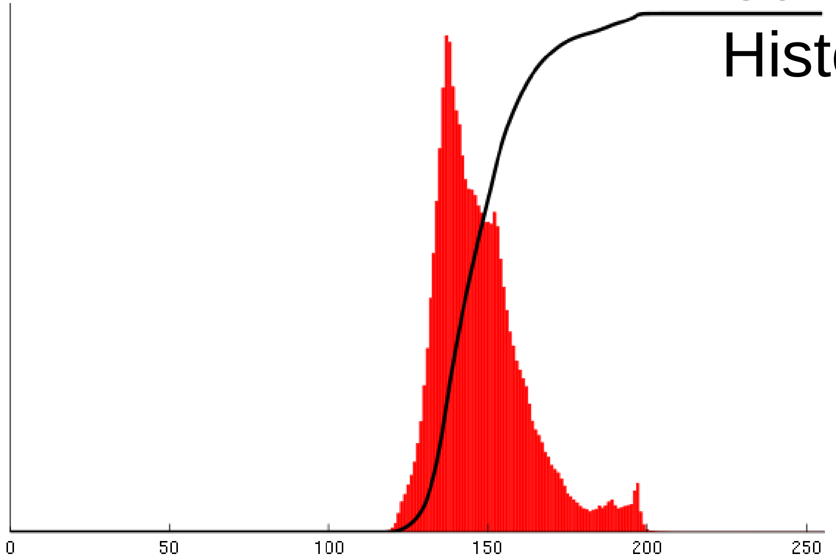
- A solution: distance between cumulative histograms $S(i) = \sum_{j=0}^{i} h(j)$



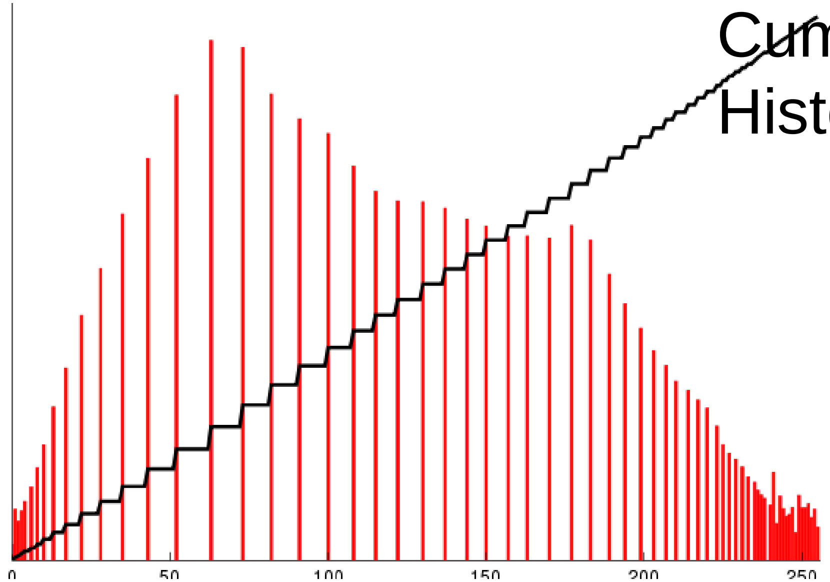
- Vector distance is now OK: $|S1 - S2| = 9$ $|S1 - S3| = 36$

Histogram Equalization Example (Wikipedia)

Original Image
& Histogram



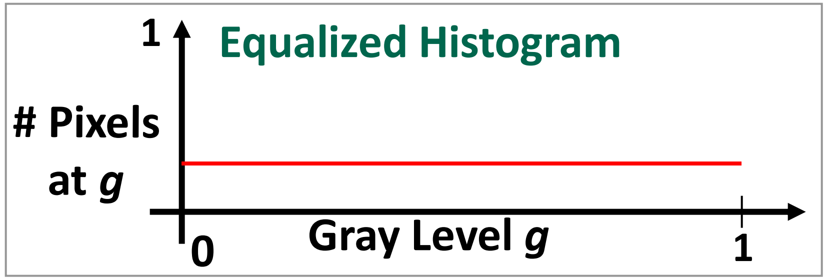
After Histogram
Equalization

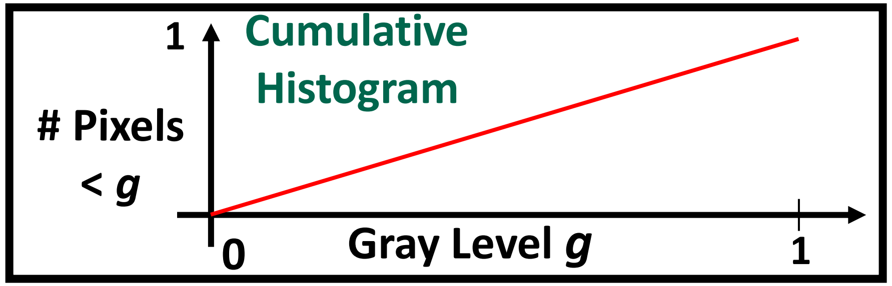


Histogram Equalization

- Equal usage of the gray level range. Integration used.

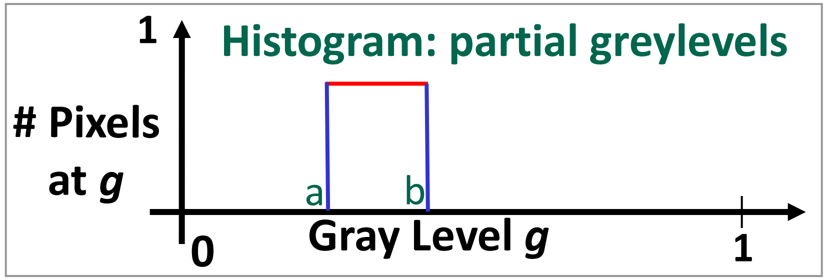
$p(g)$

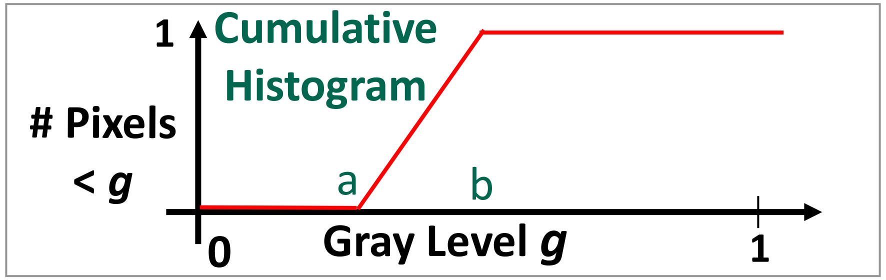




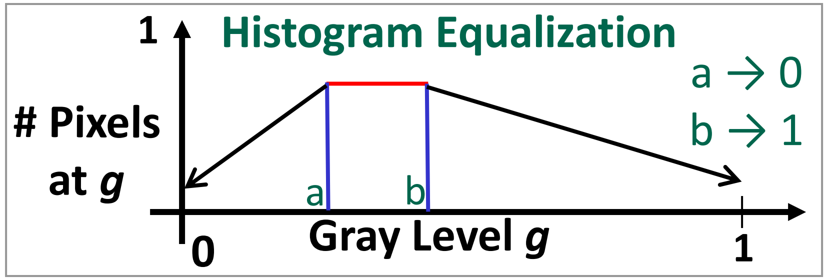
$$P(g) = \int_0^g p(x)dx$$

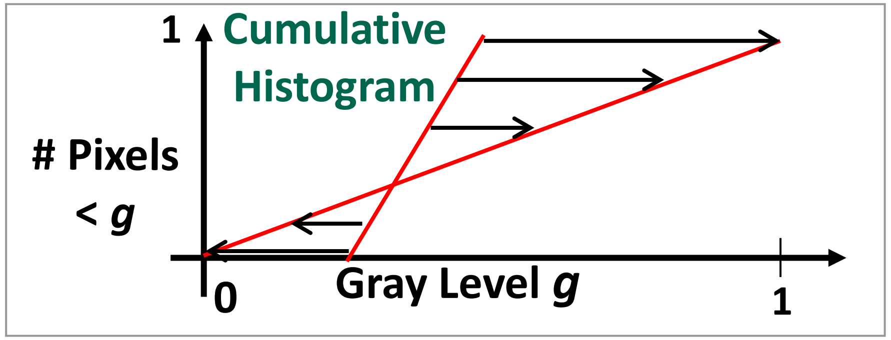
$q(g)$





$$Q(g) = \int_0^g q(x)dx$$





$$g \Rightarrow P^{-1}(Q(g))$$

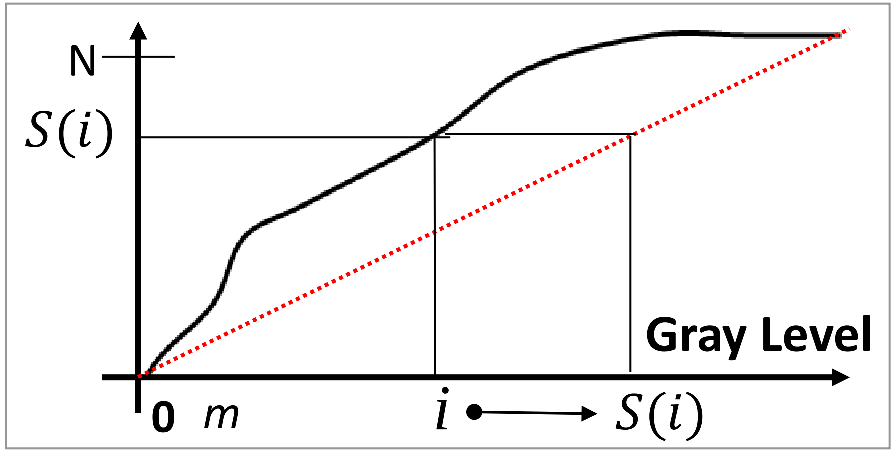
Histogram Equalization

- Compute Cumulative Histogram $S(i)$ from Histogram $h(i)$

N Pixels, grey levels $0 \,..\, K$

$h(i)$ = # pixels at grey level i

$$S(i) = \sum_{j=0}^{i} h(j)$$



1. Change every original grey level i to $S(i)$

2. <u>Stretch</u> (linear) new gray levels back to $[0 \,..\, K]$

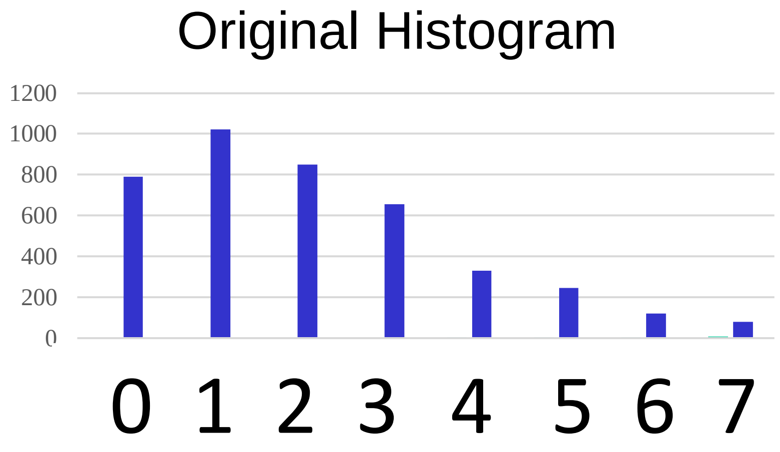
$S(m) \rightarrow 0; \quad S(q) \rightarrow K$

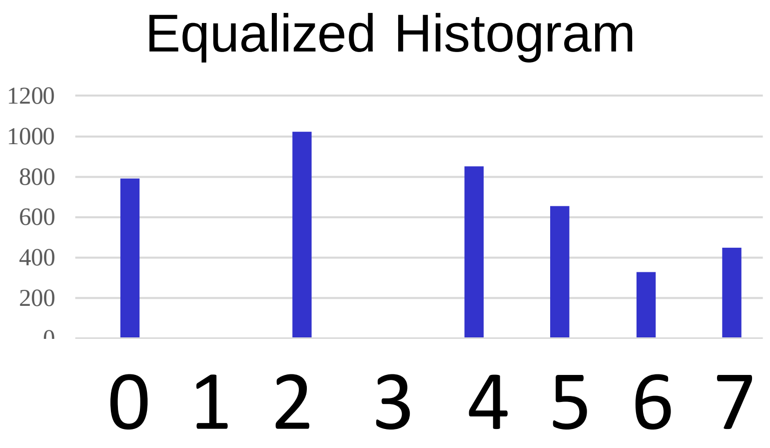
m is the lowest grey level in input image
q is the highest grey level in input image

$$i \Rightarrow K \frac{S(i) - S(m)}{S(q) - S(m)}$$

Equalization Example

Grey Level (k)	# Pixels (n)	Cumulative Histogram	Scaled 0-7	Round	New Histogram
0	790	790	0.00	0	790
1	1023	1813	2.17	2	0
2	850	2663	3.97	4	1023
3	656	3319	5.35	5	0
4	329	3648	6.05	6	850
5	245	3893	6.57	7	656
6	122	4015	6.83	7	329
7	81	4096	7.00	7	448
Total:	**4096**				**4096**





Histogram Equalization Steps

Target Range is [0..K], can be different from input range

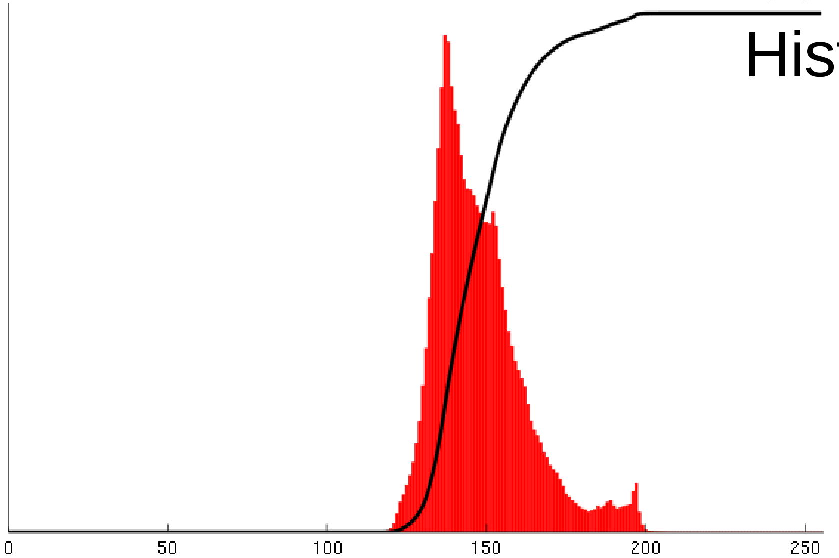
1. Given b/w image $I(x,y)$, create a histogram h:
   - For all pixels x,y: $h(I(x,y)) = h(I(x,y)) + 1$
2. [Hist] Create cumulative histogram $S(i)$:
   - $S(0) = h(0)$; $S(i+1) = S(i) + h(i+1)$;

Let $\underline{m}$ and $\underline{q}$ be the smallest & highest input grey levels

1. [Hist] Create Look Up Table (LUT) $T(i)$:
   - $T(i) = round\{K \times [S(i)\text{-}S(\underline{m})] / [S(\underline{q})\text{-}S(\underline{m})]\}$
2. Apply LUT T to image I, get equalized image J:
   - $J(x,y) = T(I(x,y))$

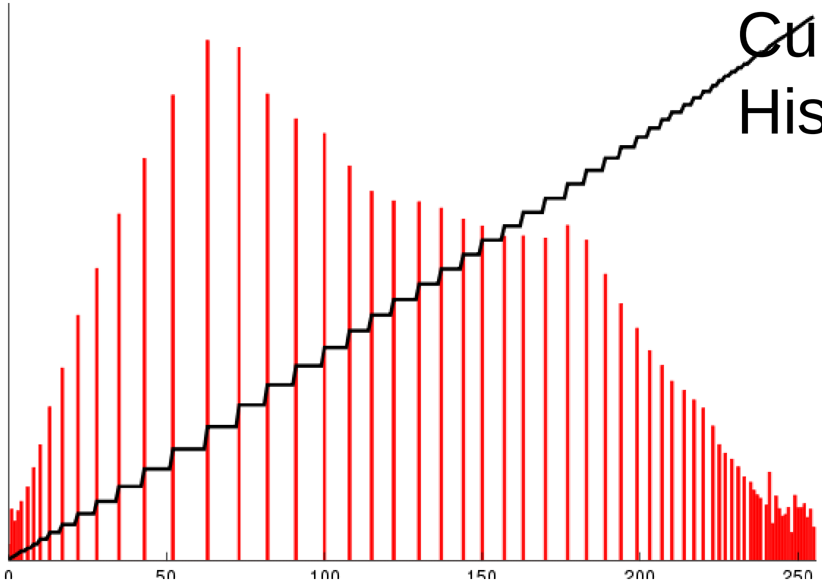
Example of Equalization (Wikipedia)

Original Image & Histogram



Cumulative Histogram

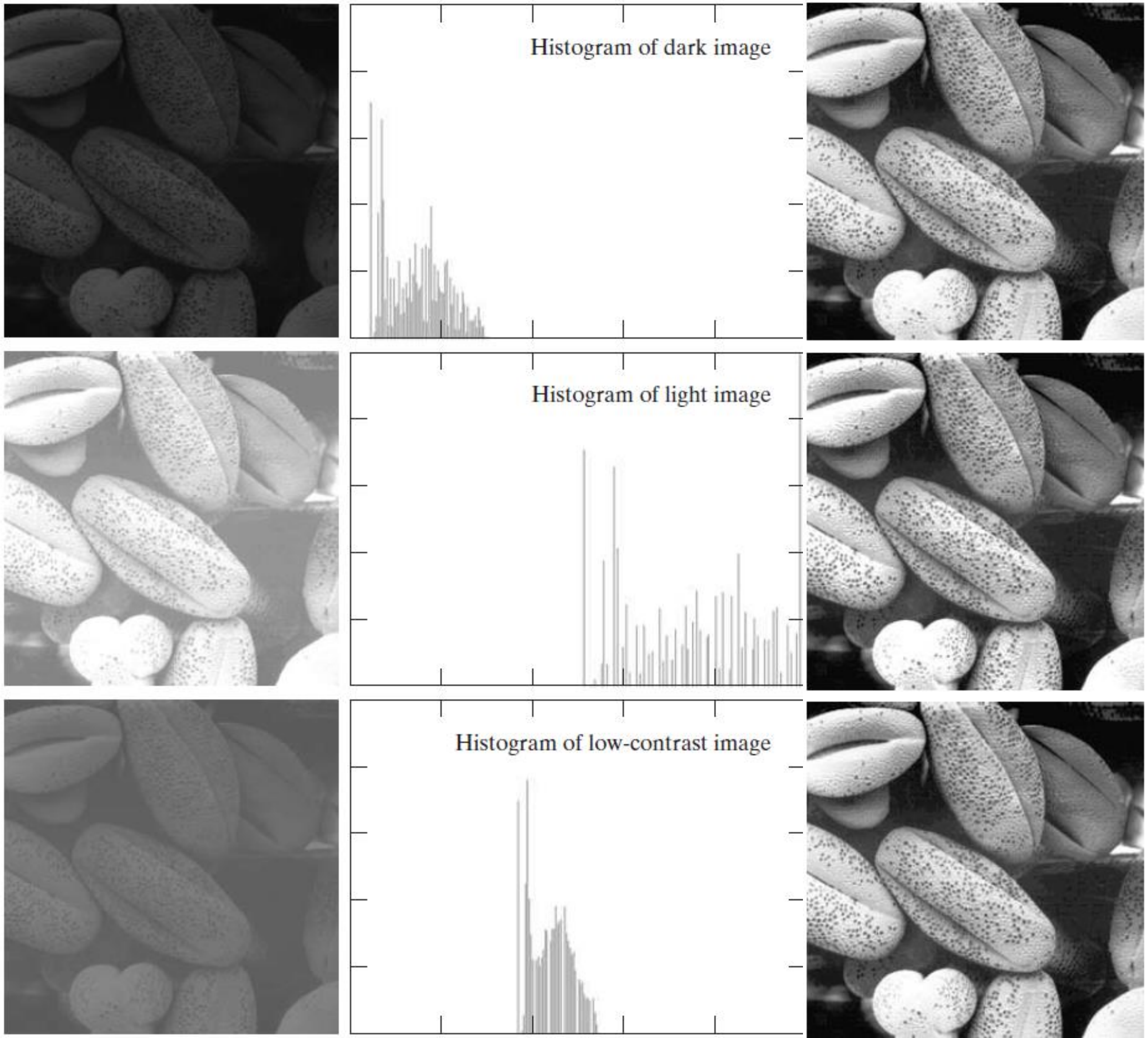
After Histogram Equalization



Cumulative Histogram

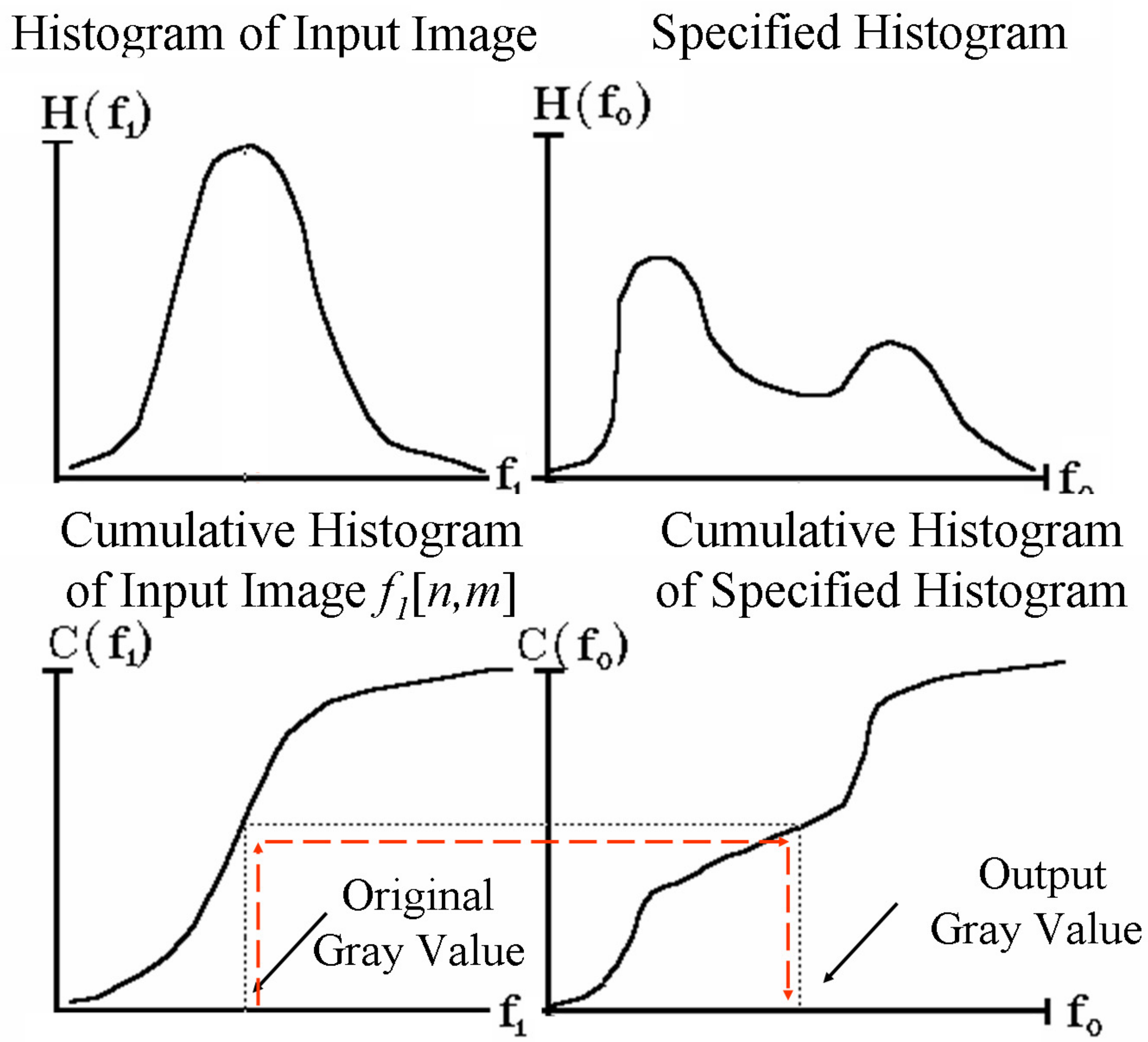
Properties of Histogram Equalization

- Monotonic Transformation: Does not reverse intensity order
- What happens if we apply Equalization twice?
- New intensity $\approx$ Cumulative Probability
  - What is the meaning of grey level 127 in an image after equalization to [0..255]?
- Will fail(?) if assumptions are not true.
  - When?



Histogram Equalization is invariant to monotonic point operations

Histogram Specification

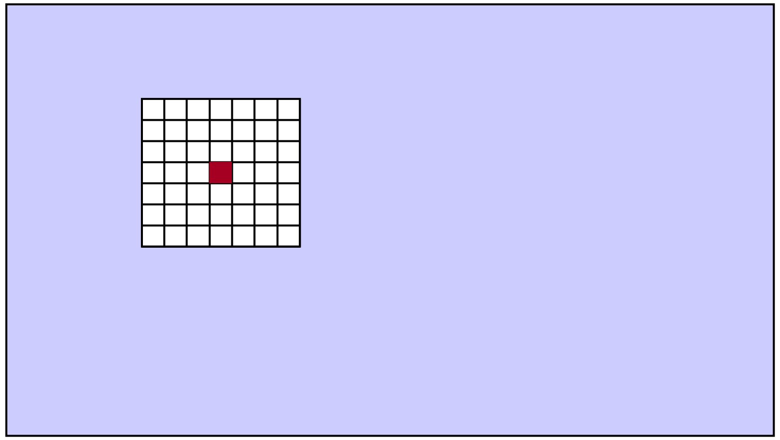


- Change the histogram of input image to any specified histogram
- Input histogram H_1 and cumulative C_1
- Target histogram H_0 and cumulative C_0
- For each input grey level u find the target grey level v such that $C_1(u)=C_0(v)$

Adaptive Histogram Equalization

- Different intensity distributions in an image
  - Example: sunny areas and shadowed areas
- Poor result for Histogram Equalization
  - Work on sunny and shadowed areas separately
  - How to segment?
- Workaround: Compute histogram in local regions around each pixel

Adaptive Histogram Equalization



- For each pixel
  - Compute Equalization LUT in local region
  - Transform by LUT only the center pixel
- Go to next pixel
- How to optimize?

Adaptive Histogram Equalization

Quantization: Reduce # of Colors

- Example: To reduce [0..255] to 32 grey levels [0..31], we use a Quantization LUT

In	0	1	...	k	...	255
Out	q_0	q_0	...	q_n	...	q_{31}

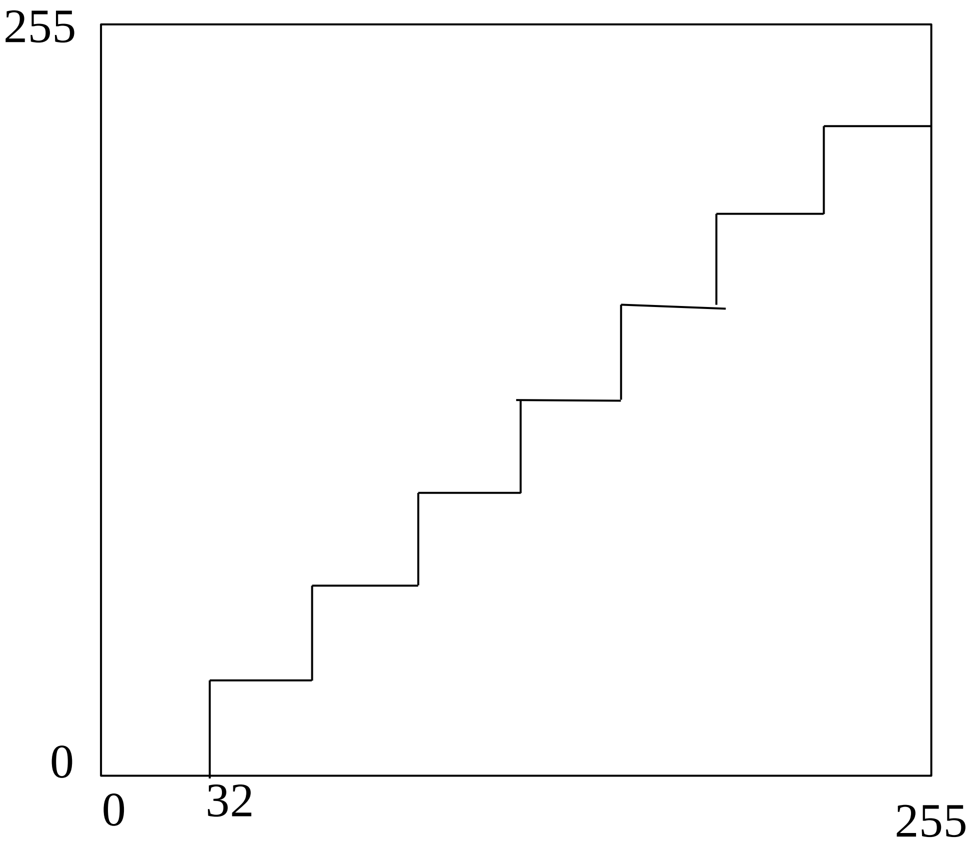
- Uniform Quantization: Every grey level *k* is mapped to *k/8*
- To restore the original image, we use a Restoration LUT

In	0	1	...	k	...	31
Out	...	G_1	...	G_k	...	G_{31}

- Example: Every *q* is mapped to *q×8* (?)
- When is uniform Quantization not optimal?

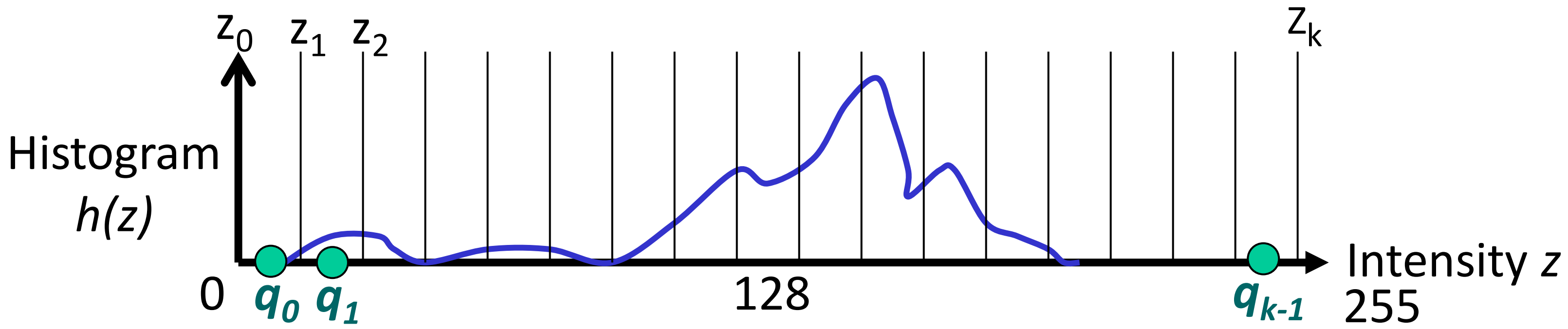
Uniform Quantization

- E.g. - Every grey level *k* is mapped to *k/8*
- To restore the original image every *q* is mapped to *q×8* (?)



0 .. 31 → 0	→ 0
32 .. 63 → 32	→ 36
64 .. 95 → 64	→ 72
96..127 → 96	→ 108
128..159 → 128	→ 142
. . .	. . .
224..255 → 224	→ 255

Quantization



- Divide grey level range into fewer segments
- Quantization: A grey level segment is mapped to one index
  - Borders of segments: $z_0=0,\ z_1,\ z_2, \ldots,\ z_k=255$
  - All grey levels in segment $[z_{i-1}, z_i]$ are mapped to $i-1$
- Restoration: Each index i will be restored to intensity q_i
- Uniform Quantization: $$z_{i+1} - z_i = (z_k - z_0)/k$$

$$q_i = (z_i + z_{i+1})/2$$

Quantization Error

- Assume that during quantization grey level ***g*** is coded by ***i***, and code ***i*** is restored as q_i
- A <u>Possible</u> quantization error **for one pixel *p*** is:

$$\mathrm{E}_p^2 = (g - q_i)^2$$

- The total error introduced by quantization of all pixels is:

$$\mathrm{E}^2 = \sum_{pixels\ p} \mathrm{E}_p^2 \qquad = \sum_{i=0}^{255} hist(i)\mathrm{E}_i^2$$

All pixels with same grey will level will have same quantization error

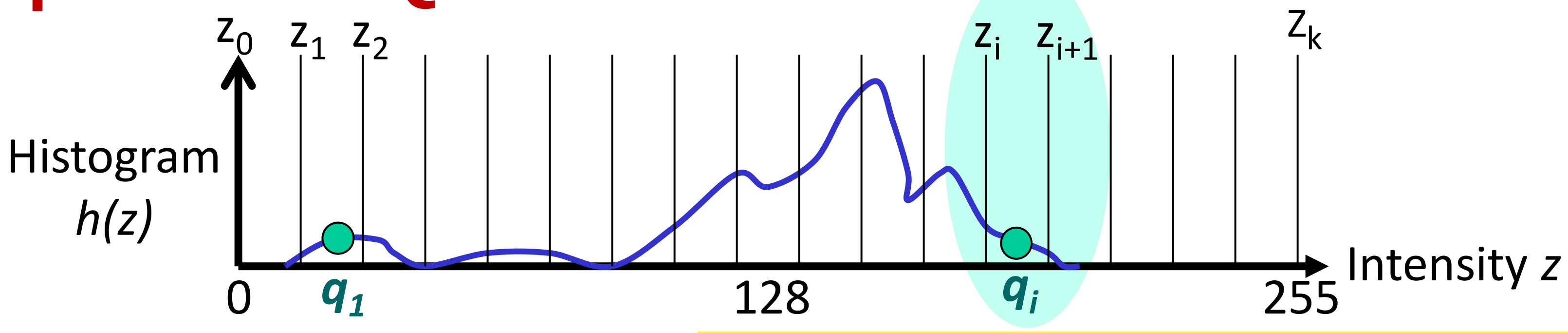
- Unknown optimal transformations: $g \rightarrow i$, $i \rightarrow q_i$

SSD As a Quantization Error

- SSD (L_2, Sum of Squared Differences) is a poor error measure compared to human perception
- There are suggestions for other measures, but they are harder to compute and to analyze (E.g. L_1, Sum of Absolute Values)
- $SSD = 0$ implies the same image…

- (Best way to compute image similarity is with Neural networks…)

Optimal Quantization - Condition



- Minimize the error:

$$\sum_{i=0}^{k-1}\left(\sum_{g=\lfloor z_i\rfloor+1}^{\lfloor z_{i+1}\rfloor}(q_i-g)^2h(g)\right)$$

- Solution (Prove!):

$$q_i=\frac{\sum_{g=\lfloor z_i\rfloor+1}^{\lfloor z_{i+1}\rfloor} g\cdot h(g)}{\sum_{g=\lfloor z_i\rfloor+1}^{\lfloor z_{i+1}\rfloor} h(g)} \qquad z_i=\frac{q_{i-1}+q_i}{2}$$

Optimal Quantization - Process

- Find z_i and q_i such that

$$q_i = \frac{\sum_{g=\lfloor z_i \rfloor+1}^{\lfloor z_{i+1} \rfloor} g \cdot h(g)}{\sum_{g=\lfloor z_i \rfloor+1}^{\lfloor z_{i+1} \rfloor} h(g)} \qquad z_i = \frac{q_{i-1} + q_i}{2}$$

- This is done iteratively
- Initial guess: find z_i such that for all i there are same number of pixels whose grey level is between z_i and z_{i+1}.
- From initial guess of z_i compute q_i .
- Iterate: computing z_i and than q_i until convergence.

Next: Fourier

- Discrete Fourier Transform in Wikipedia
- http://homepages.inf.ed.ac.uk/rbf/HIPR2/fourier.htm
- https://betterexplained.com/articles/an-interactive-guide-to-the-fourier-transform/