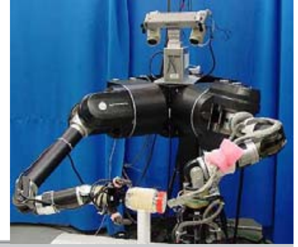
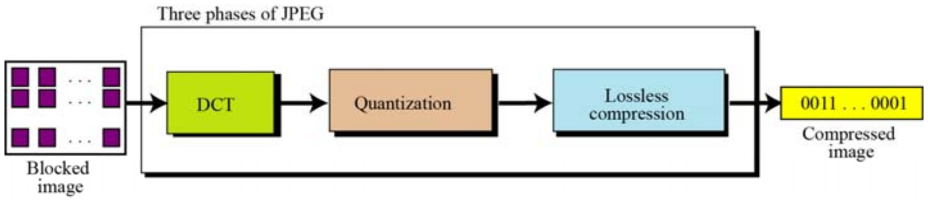
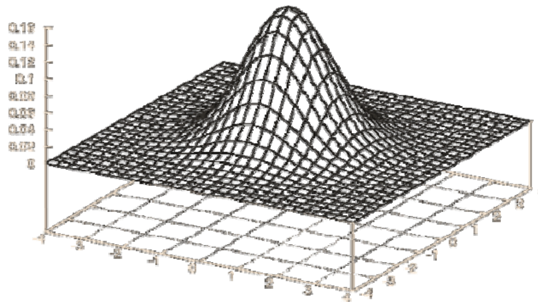
EVC - Computer Vision
Rehersal 1

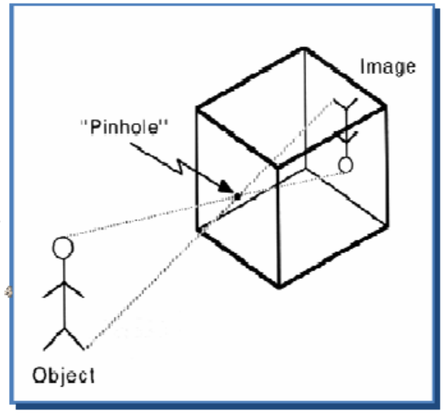


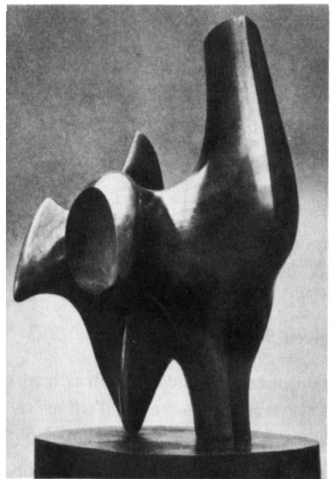
http://www.caa.tuwien.ac.at/cvl/teaching/sommersemester/evc

- Content:
  - Image Acquisition
  - Image Encoding and Compression
  - Point Operations
  - Local Operations
  - Image Sensors
  - Edge Filtering

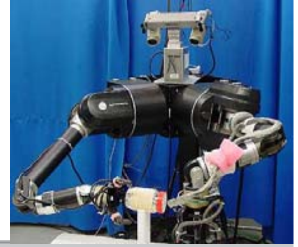


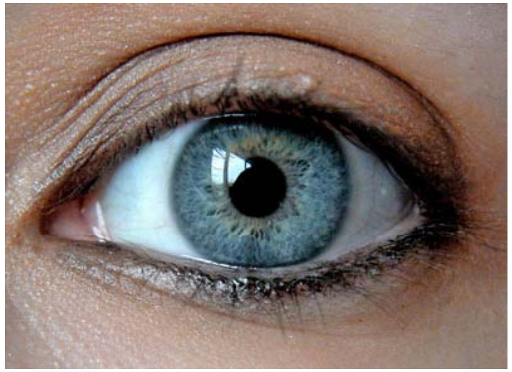


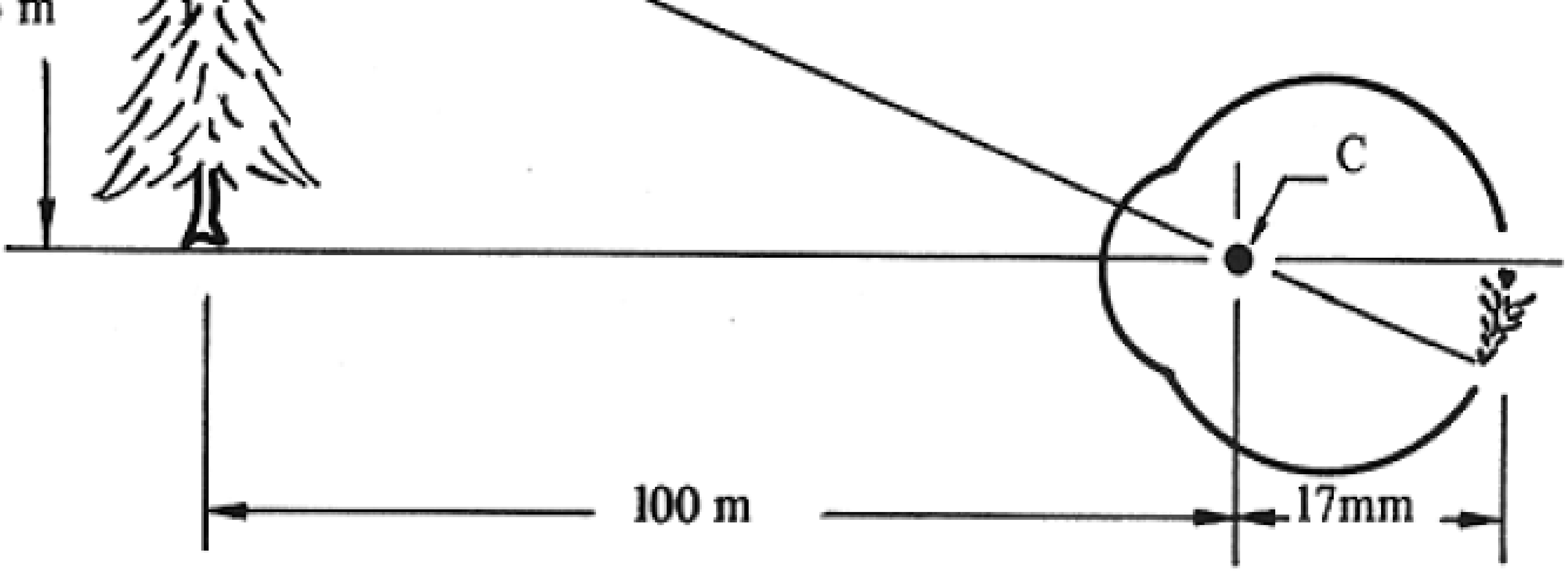




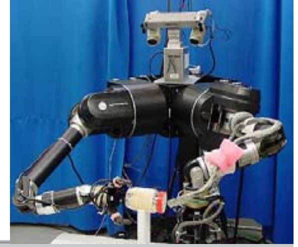
Human Eye



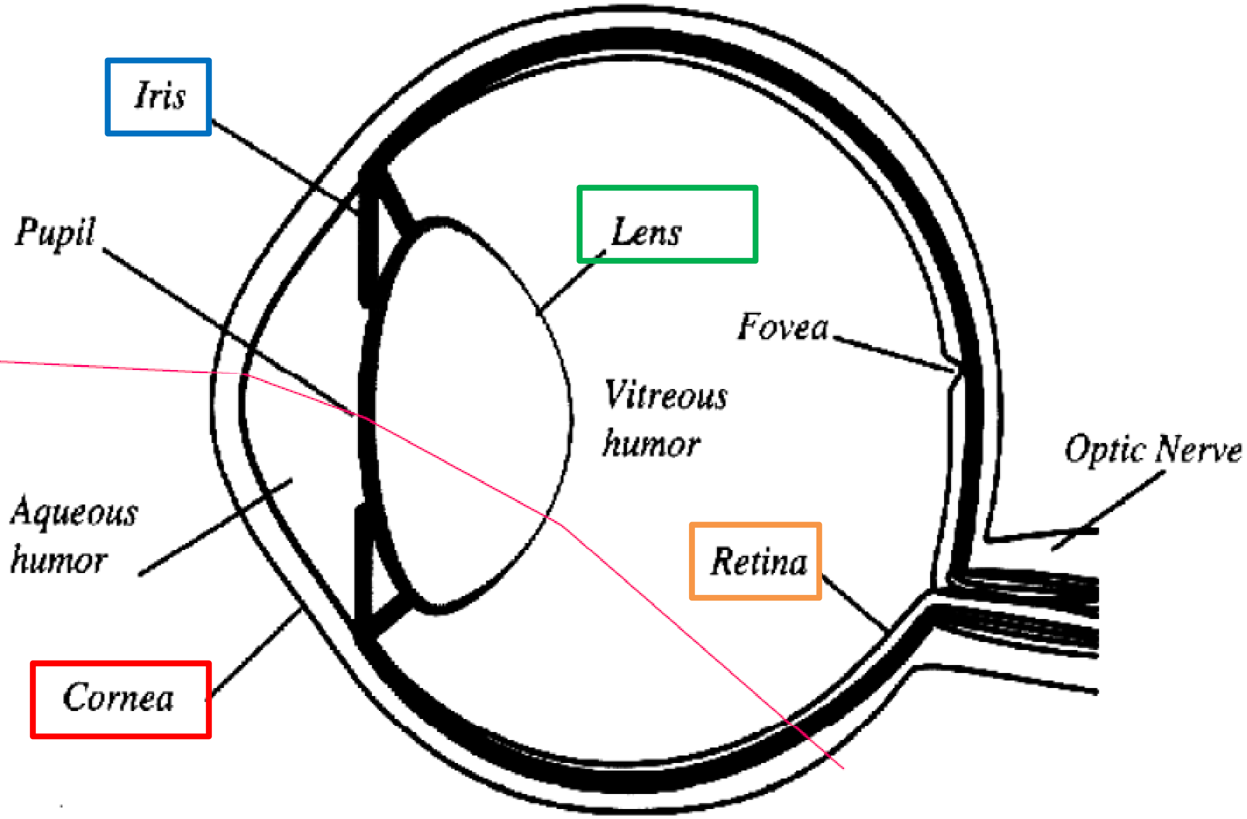




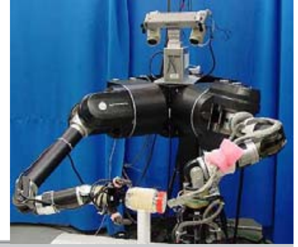
Human Eye - Components



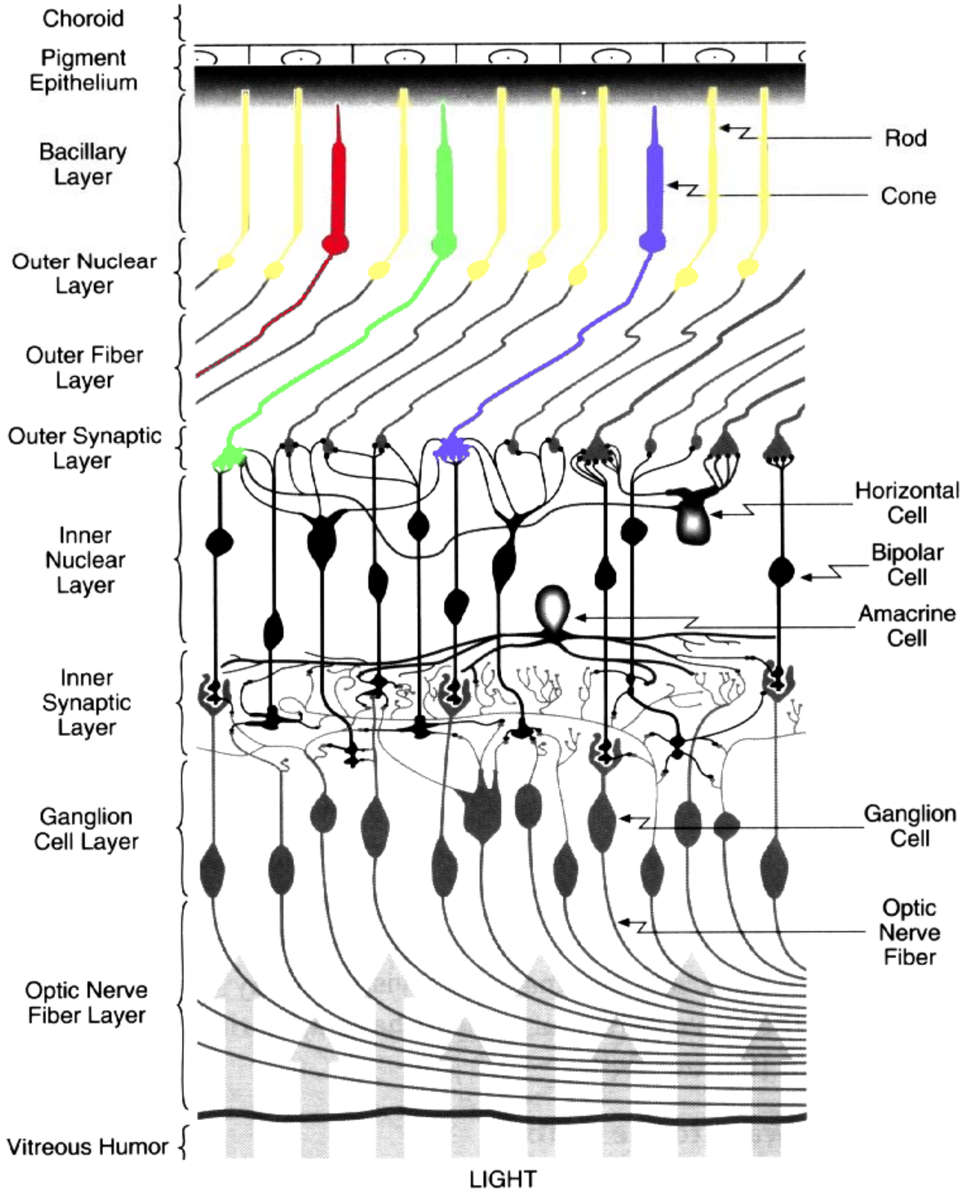
- Cornea + Lens:
  - Light fraction
- Iris:
  - variable aperture
- Retina: Image Detector
  - (ca. 100 Mio. Photoreceptors)



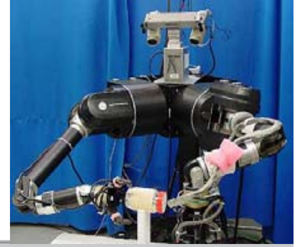
Retina



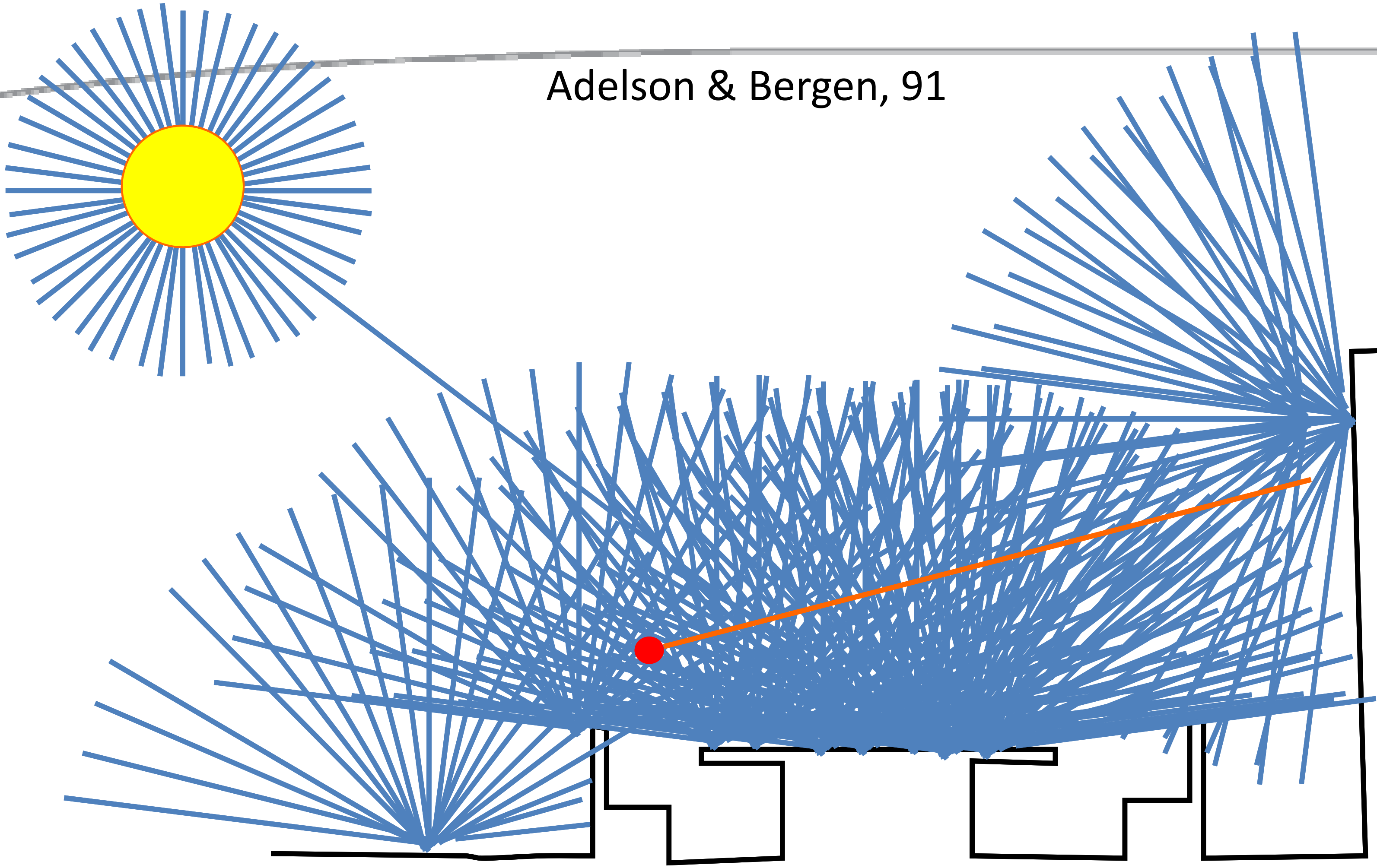
- Rods: Monochrome
- Cones: Color (RGB)
- Fovea: Cones only
- Number: 6 Mio. Cones
  120 Mio. Rods
- But only **1 Mio. nerve fibers** in optic nerve => **intelligent sensor**!



The Plenoptic Function



Adelson & Bergen, 91



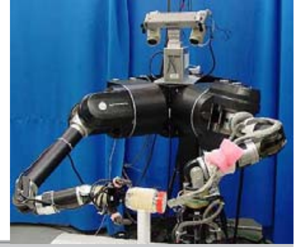
The intensity P can be parameterized as:

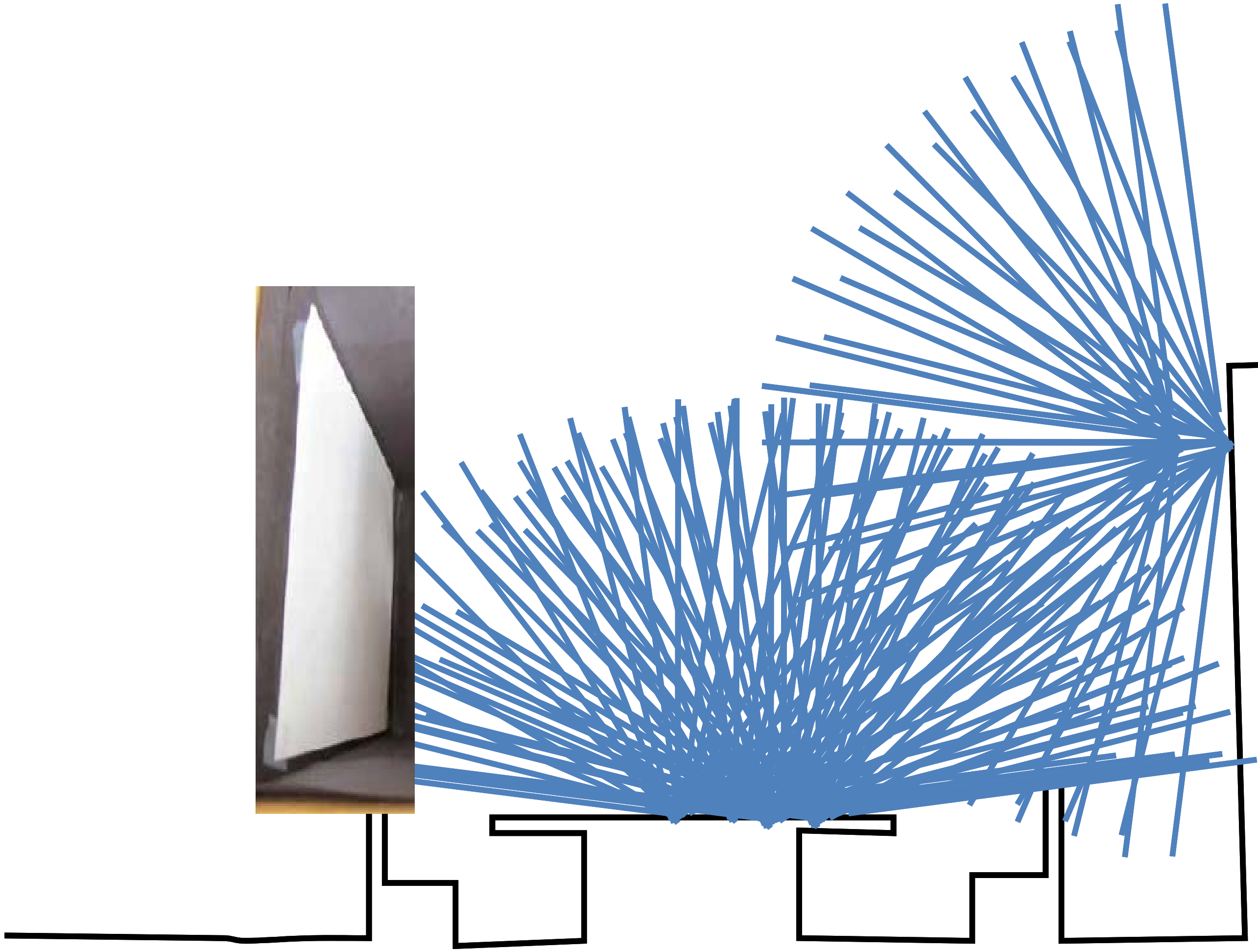
$$P\ (\theta,\ \phi,\ \lambda,\ t,\ Vx,\ Vy,\ Vz)$$

Image coordinates (sperical)
Color
Time
3D space

"The complete set of all convergence points constitutes the permanent possibilities of vision."
Gibson

Measuring the Plenoptic Function





Why is there no picture appearing on the paper?

Measuring the Plenoptic Function

The camera obscura
The pinhole camera

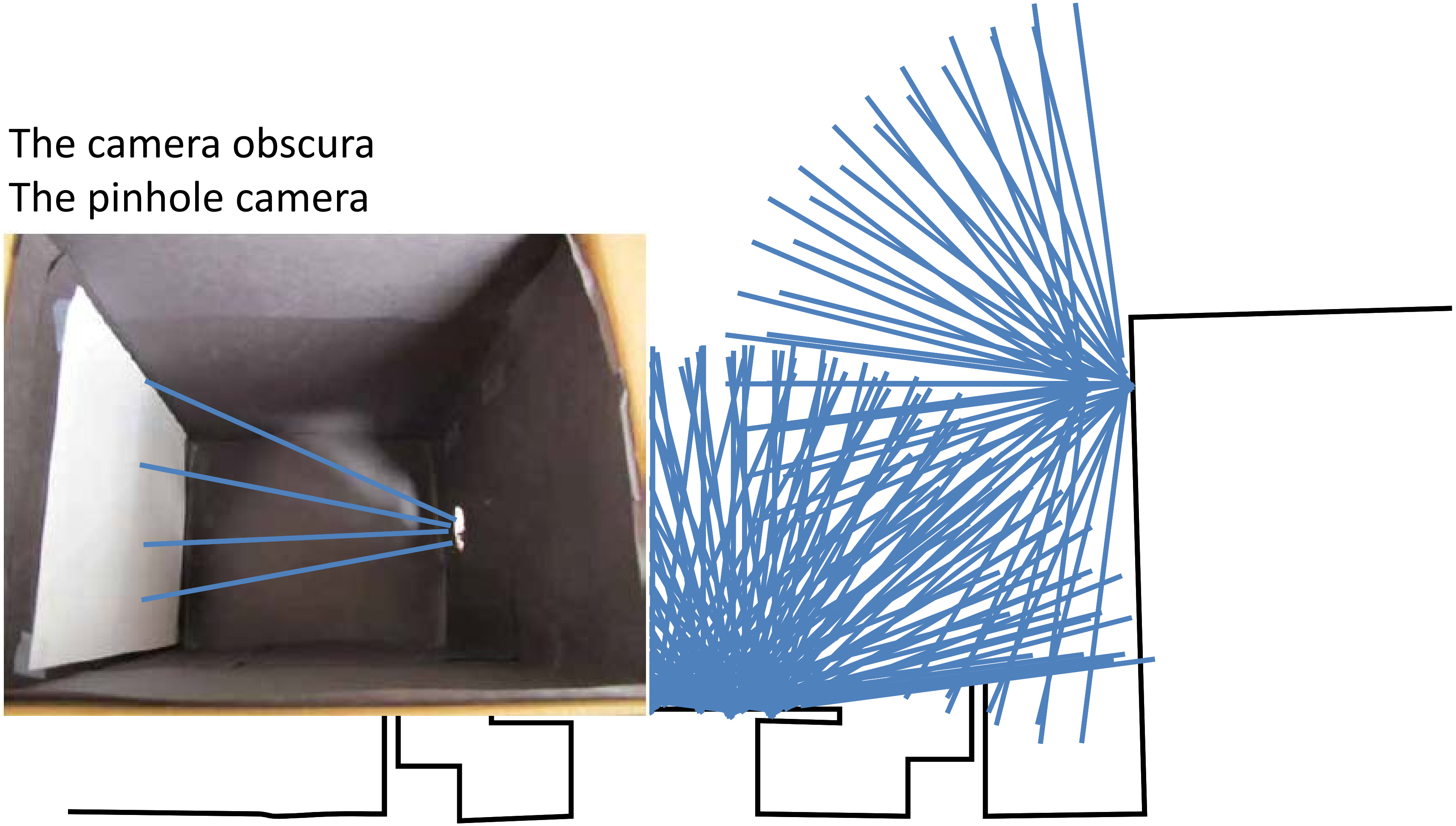
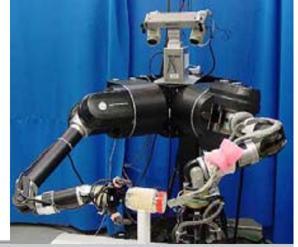


Image Geometry



- Simplest Model: Pinhole camera
  - Has a very small hole (Aperture = ∞), Light is led through the hole and forms an image at the back of the box (upside down and side-inverted)

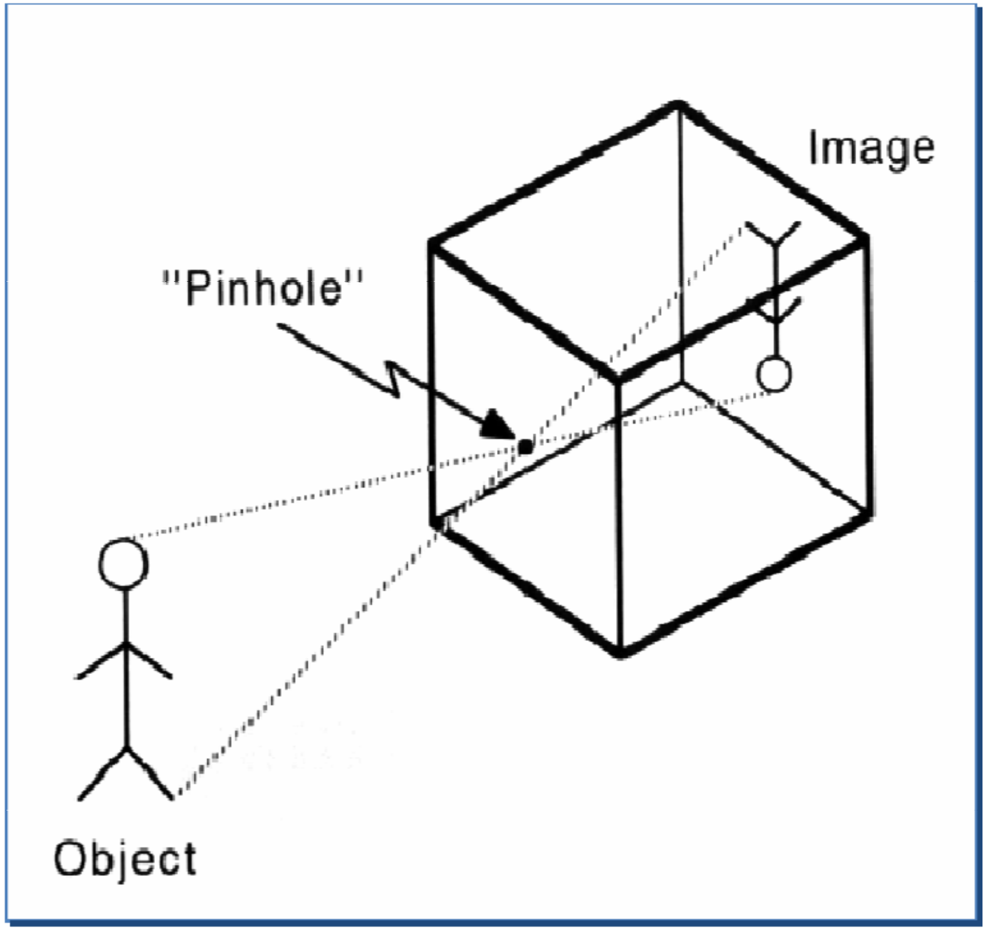
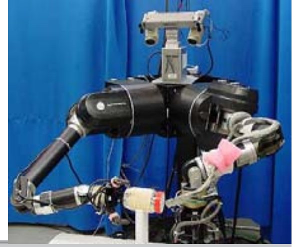
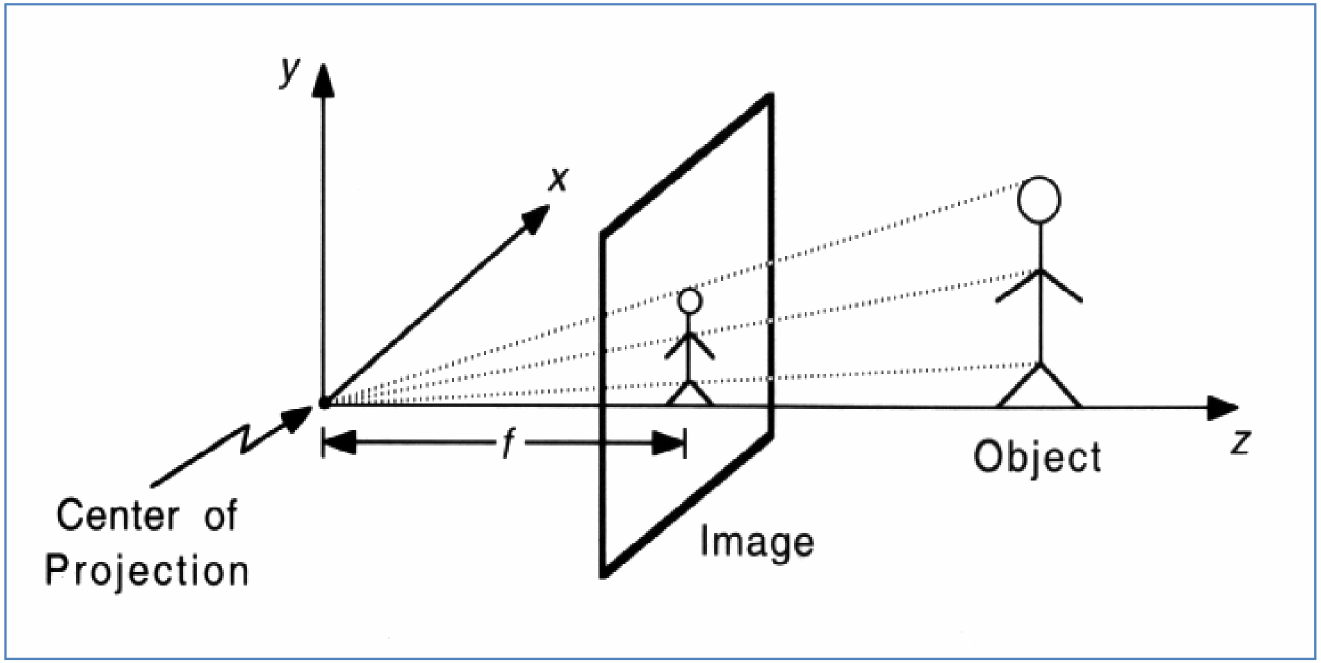


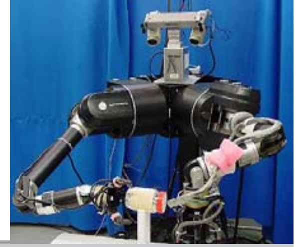
Image Geometry



- Perspective Projection (Central projection)
  - Is the projection of the 3d world onto a 2d plane by rays passing through a common point the center of projection.
  - => models image formation by a pinhole camera

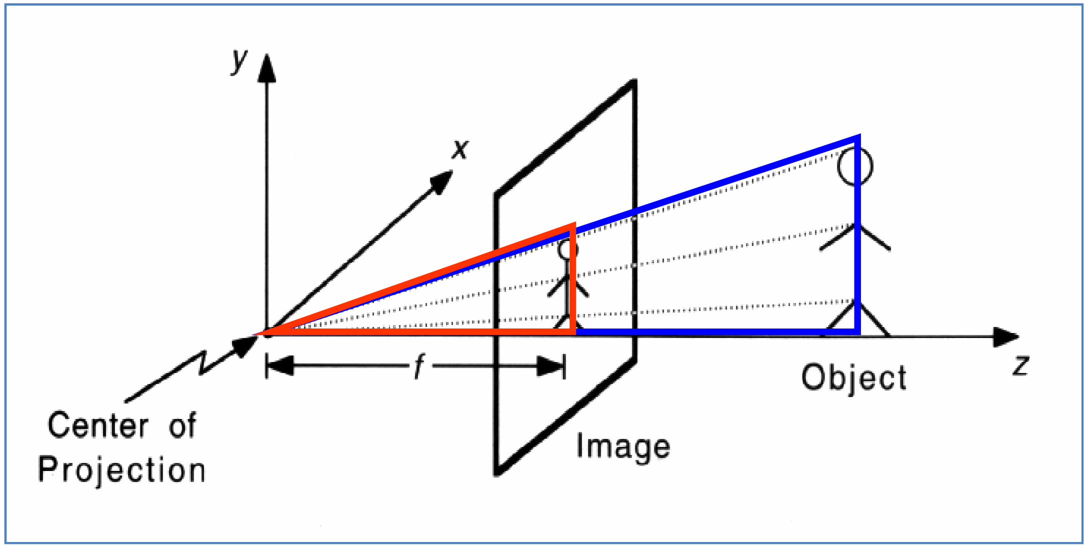


Equations of the perspective projection



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

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

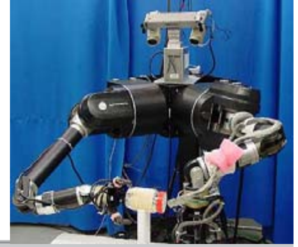


$$\frac{x}{X} = \frac{f}{Z}$$

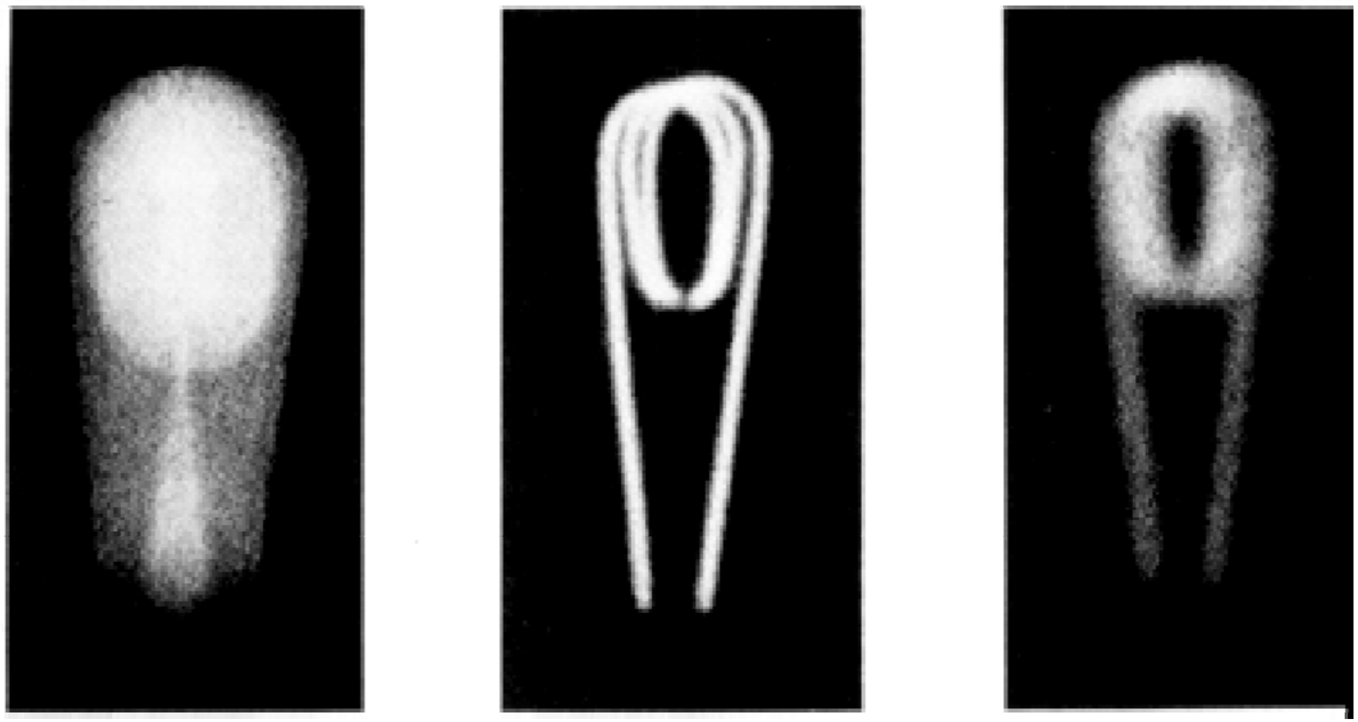
$$\frac{y}{Y} = \frac{f}{Z}$$

- Perspective projection is non-linear !

Recap: Limits of Pinhole Cameras

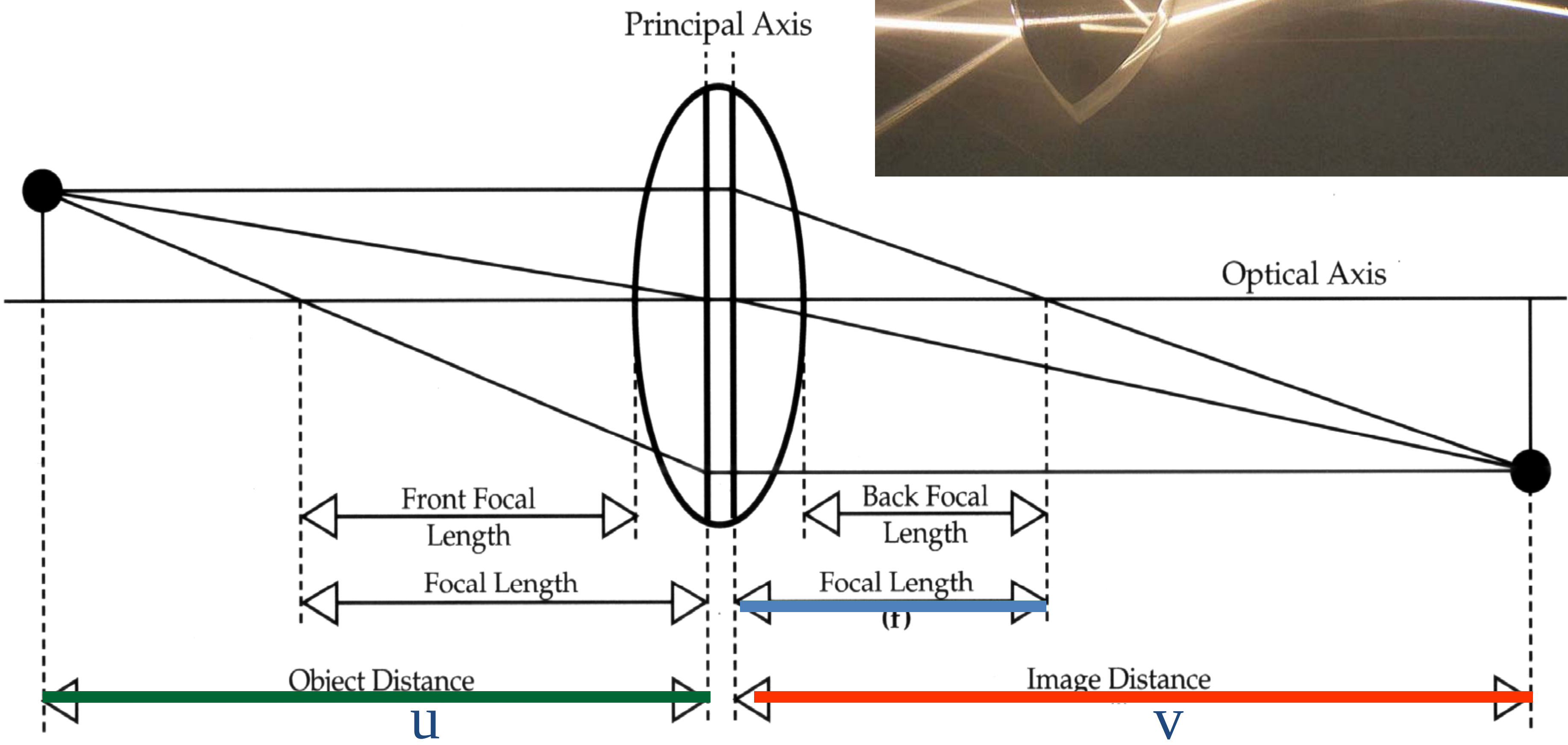


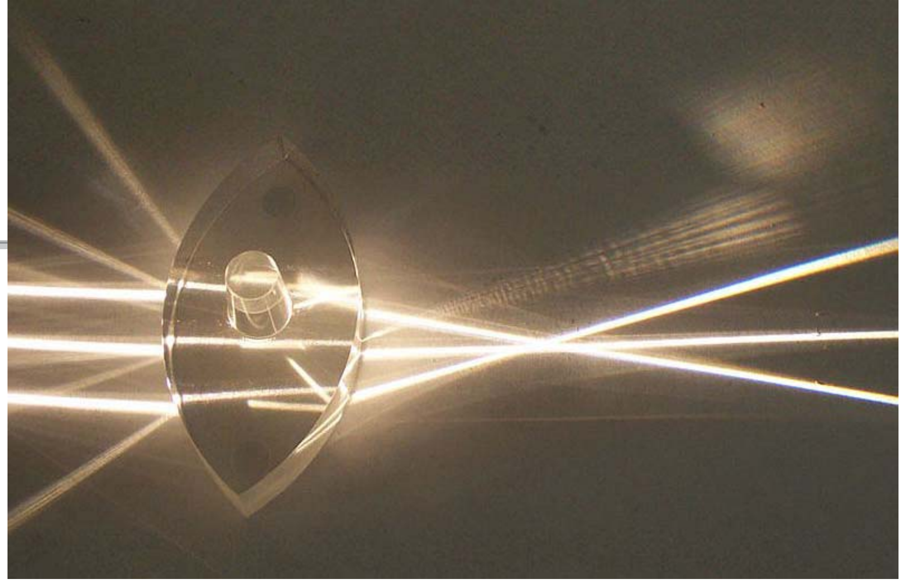
- A picture of a filament taken with a pinhole camera. In the image on the left, the hole was too big (blurring), and in the image on the right, the hole was too small (diffraction).



Ruechardt, 1958

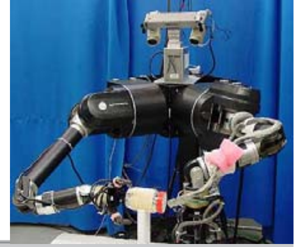
Simple Lens Parameters





FOCAL LENGTH = Distance from focus point to principal axis.

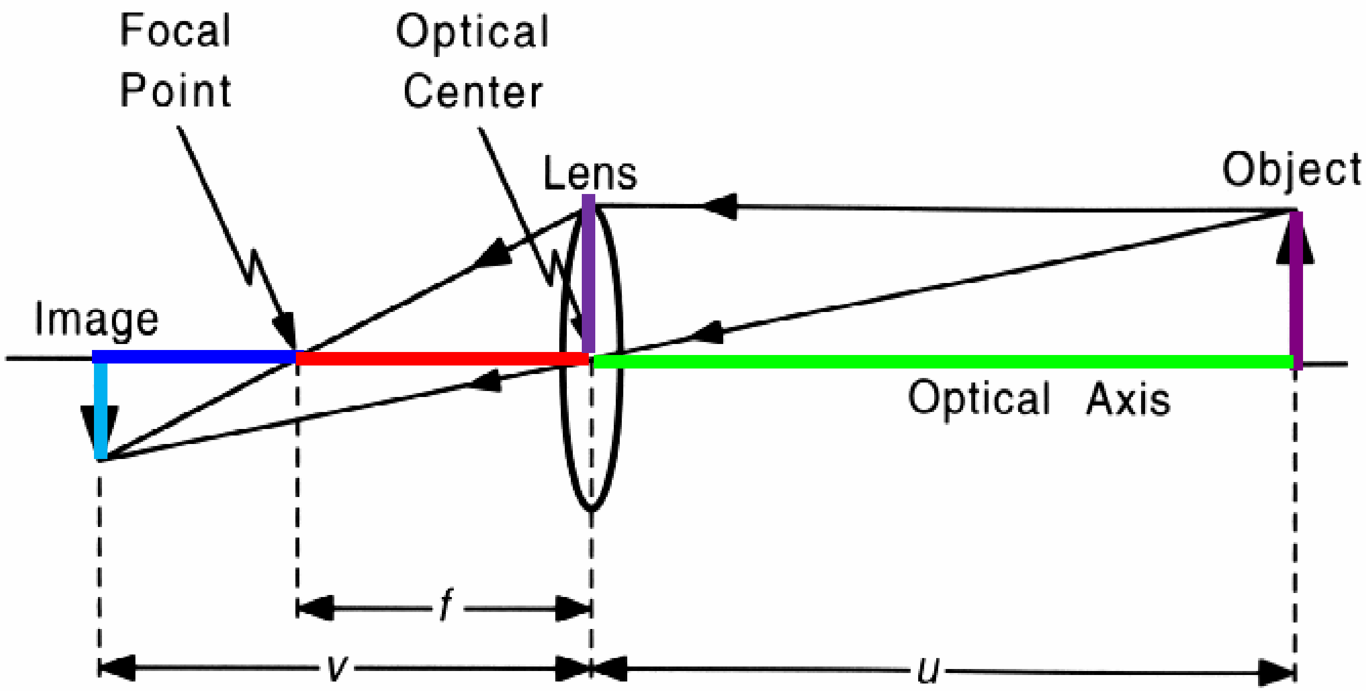
Lenses



- Pin has no lens => small Aperture => few light
  - „thin" lenses: small Aperture but much light
- Thin lens law:

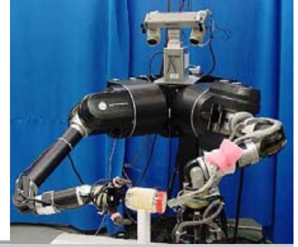
$$\frac{y_0}{y_i} = \frac{u}{v}$$

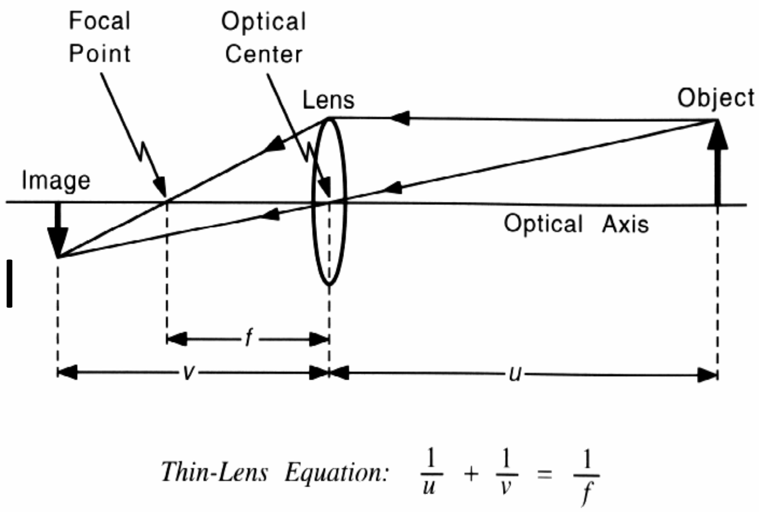
$$\frac{y_0}{y_i} = \frac{f}{v - f}$$



Thin-Lens Equation: $\frac{1}{u} + \frac{1}{v} = \frac{1}{f}$

Lenses

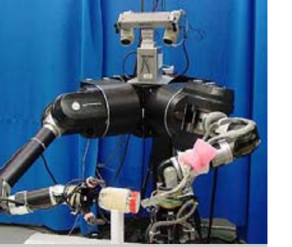




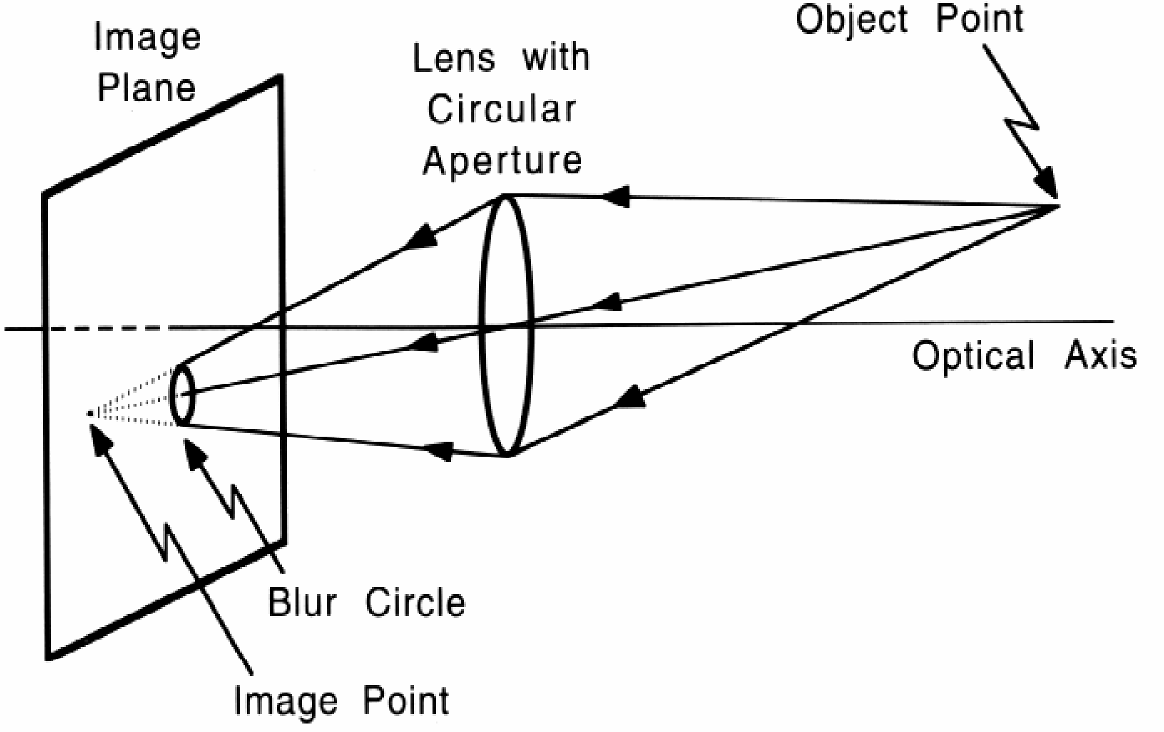
- f: focal length = distance of the point on the optical axis where all rays emerging from infinity meet to the lens plane (= all rays are parallel to the optical axis)
- if u = ∞ then v = f
- Rays going through the optical center of the lens are not diffracted
- Field of view: area that is recorded by a camera:
  - The bigger f the smaller the area that is imaged
  - Wide-angle - small f; Zoom - large f

$$\frac{1}{u} + \frac{1}{v} = \frac{1}{f}$$

Depth of Field

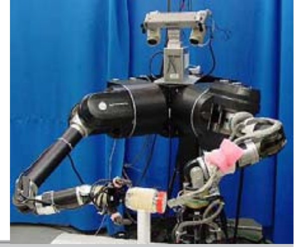


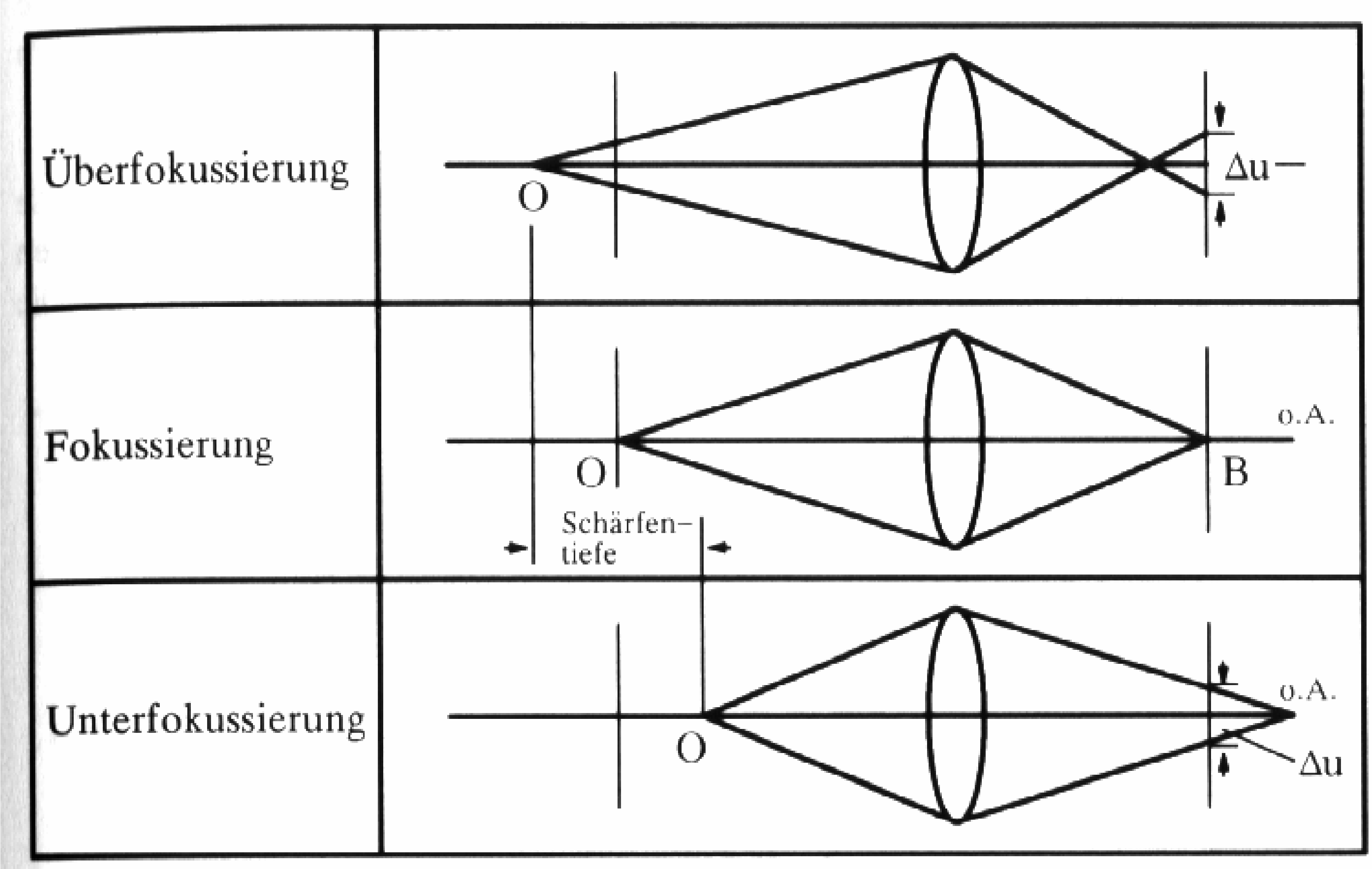
- Only objects in a certain distance are imaged sharply at the image plane, all other distances are blurred because of blur circles.
- The bigger the aperture, the bigger the blur circles
- The smaller the aperture, the sharper is the image



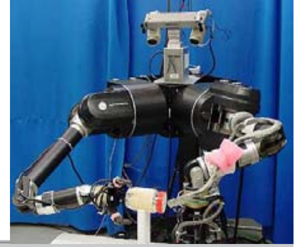
- ☐ The bigger the depth of field the darker the image
- ☐ Large Aperture = small depth of field

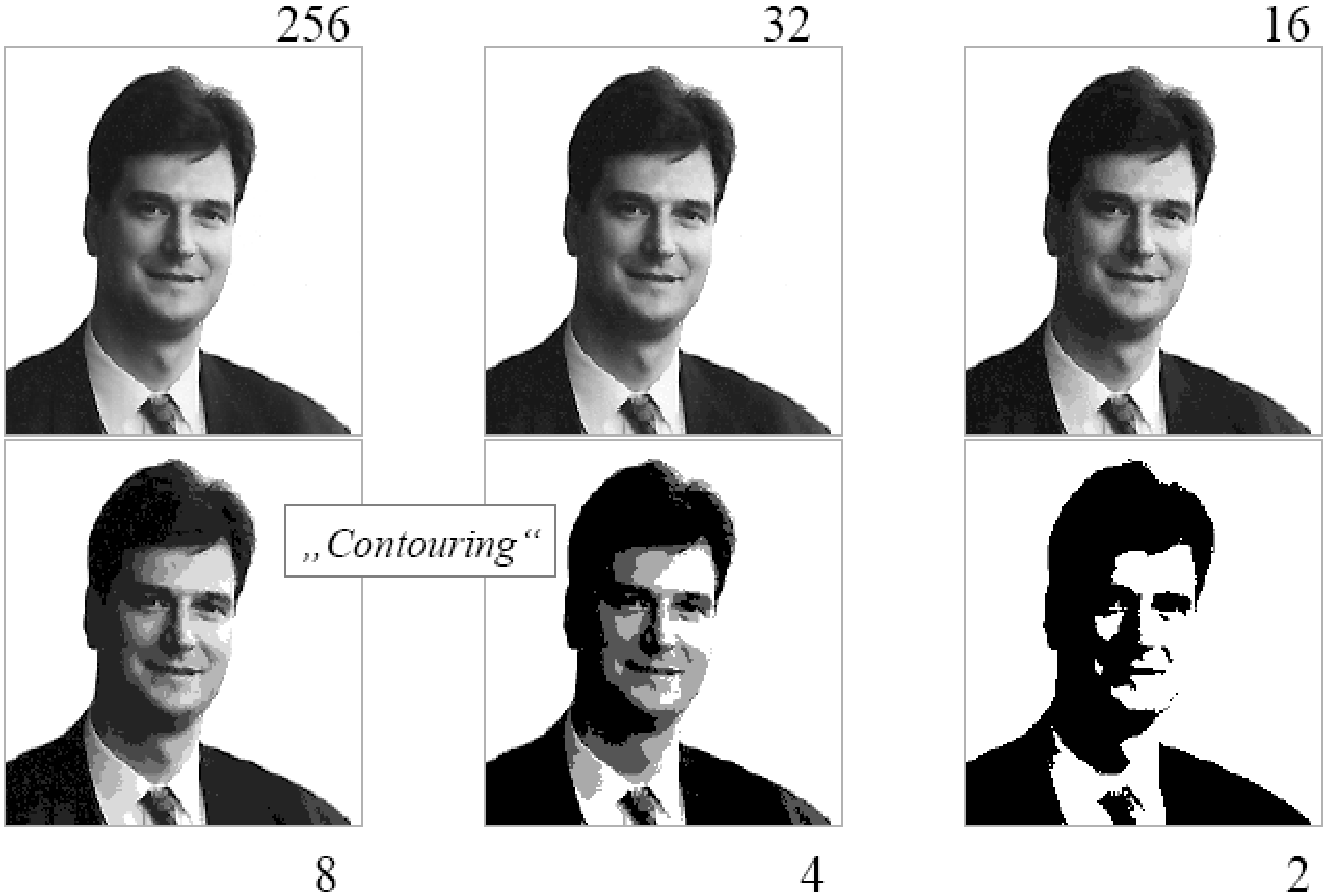
Depth of Field



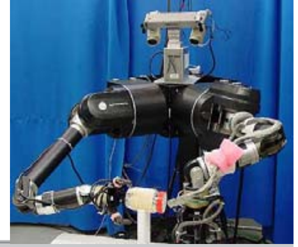


Different numbers of Gray Levels





Radiometric Resolution



- Number of digital values (“gray levels”) that a sensor can use to express variability of signal (“brightness”) within the data
- Determines the information content of the image
- The more digital values, the more detail can be expressed
- Determined by the number of bits of within which the digital information is encoded

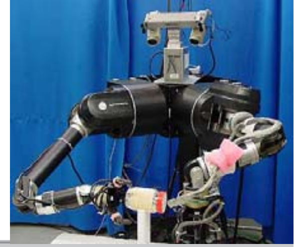
2^1 = 2 levels (0,1)

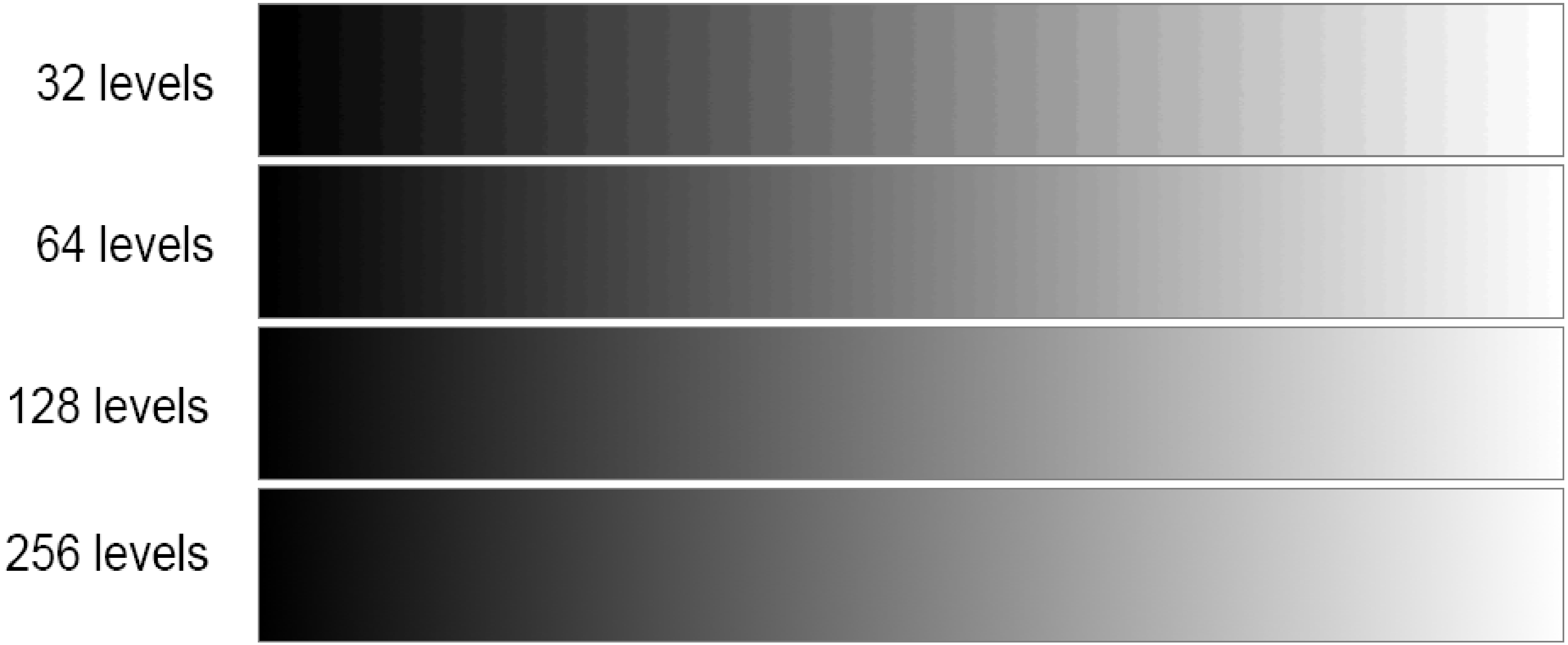
2^2 = 4 levels (0,1,2,3)

2^8 = 256 levels (0-255)

2^{12} = 4096 levels (0-4095)

How many gray levels are required?

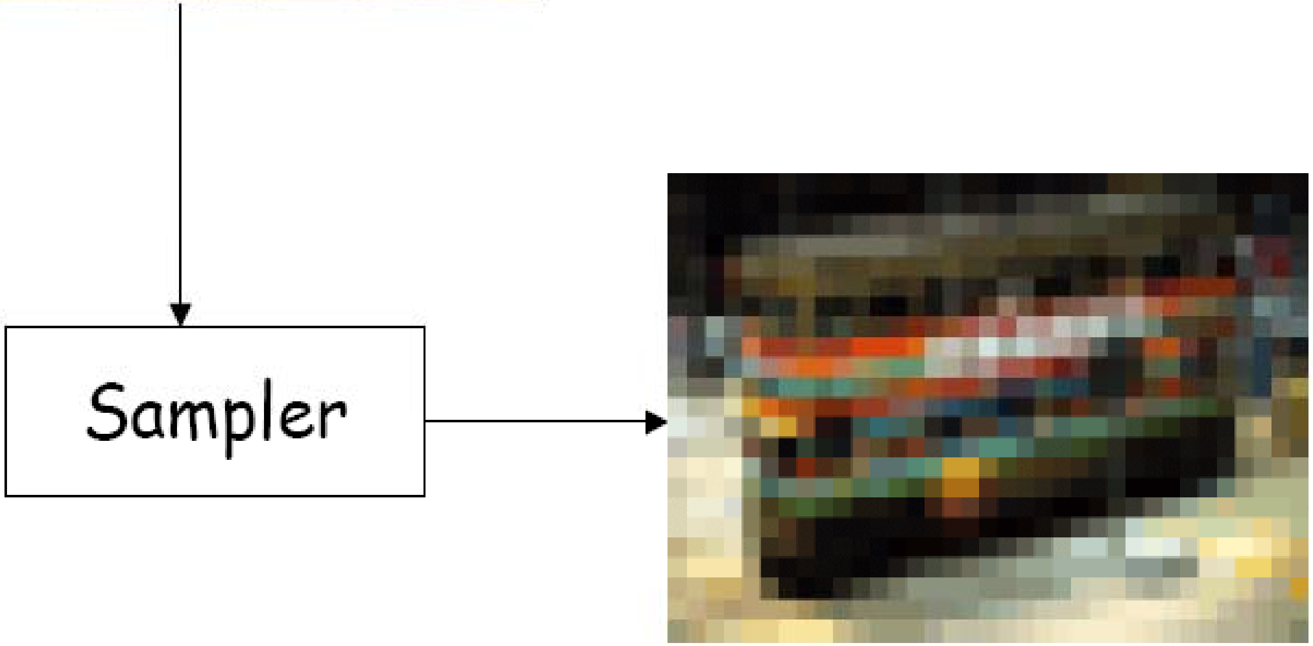




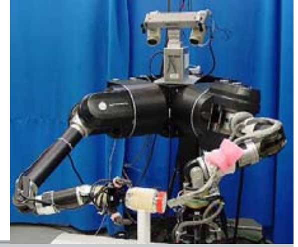
- Contouring is most visible for a ramp
- Digital images typically are quantized to 256 gray levels.

Transition to a Digital Image - 1

1. Räumliche Abtastung
(Sampling)



Transition to a Digital Image - 2



2. Diskretisierung der Bildwerte (Quantisierung)

Quantizer →

50	23	7	9	
19	8	4		
6	10			

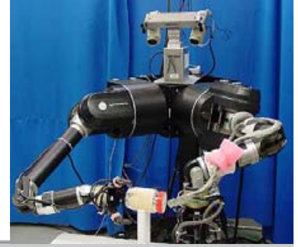
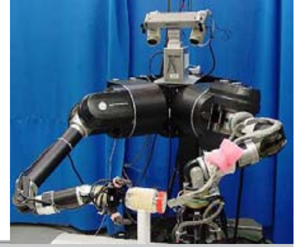


Image Size and Resolution

- These images were produced by simply picking every n-th sample horizontally and vertically and replicating that value nxn times:

200x200 100x100 50x50 25x25

Sampling Theorem



Shannon Theorem: Exact reconstruction of a continuous-time baseband signal from its samples is possible if the signal is bandlimited and the **sampling frequency** is **greater than twice the signal bandwidth**.

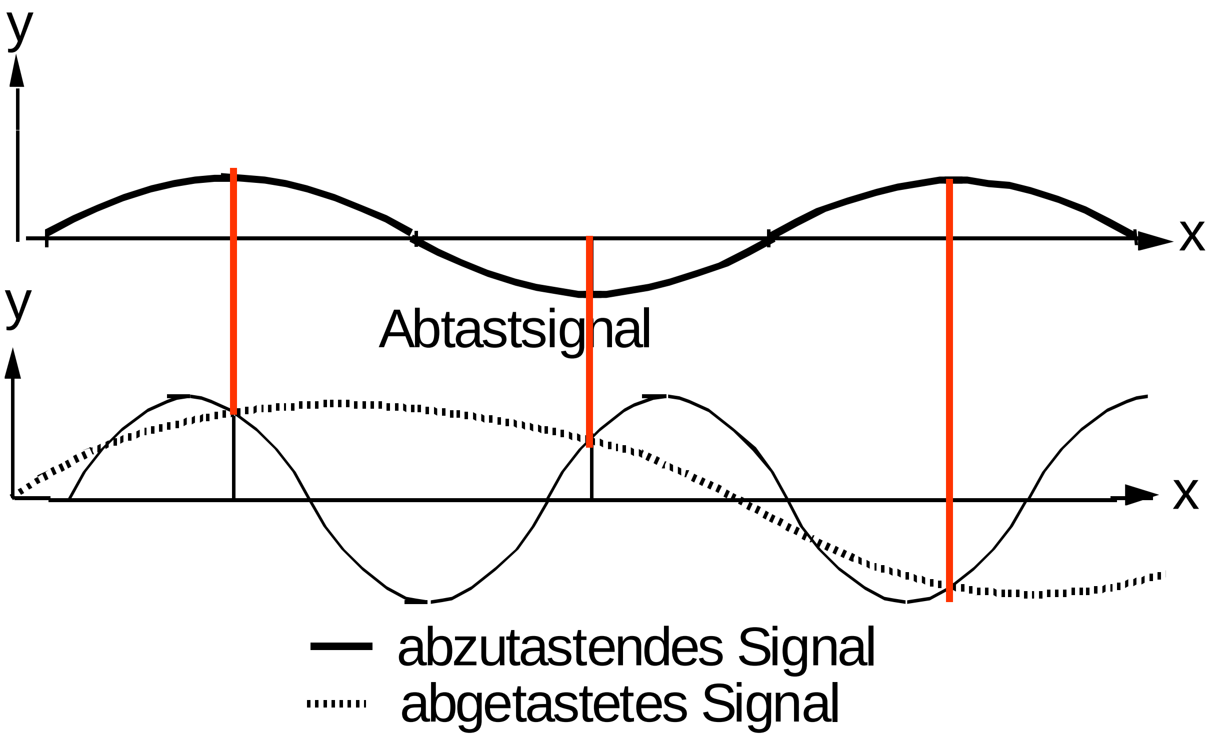
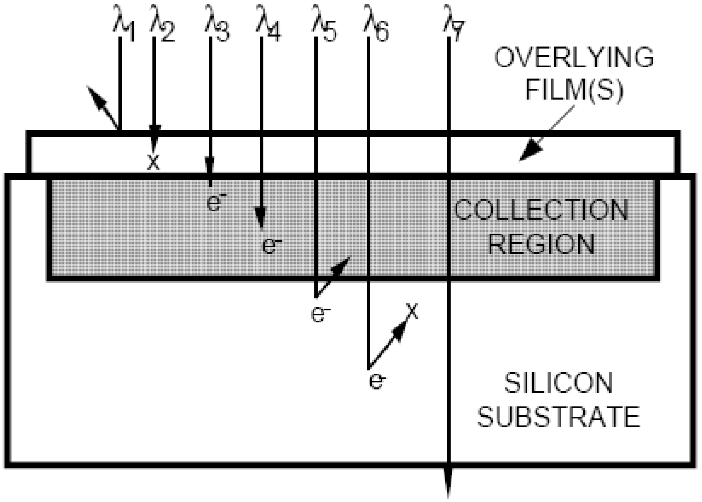
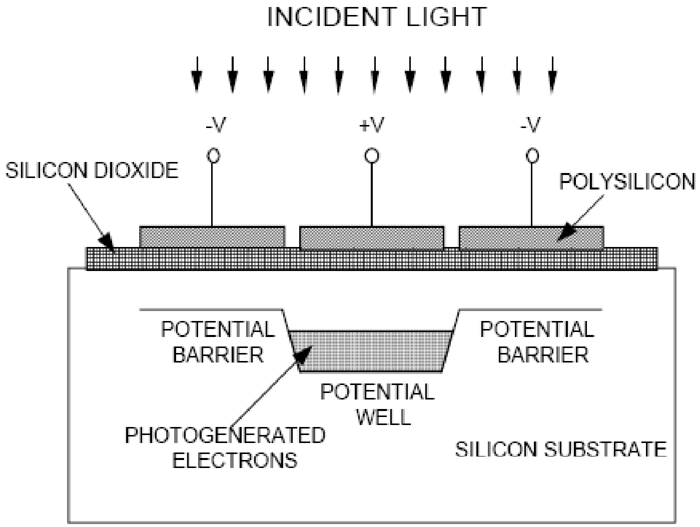


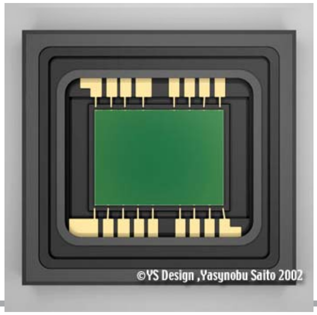
Image Sensors

- Convert light into electric charge

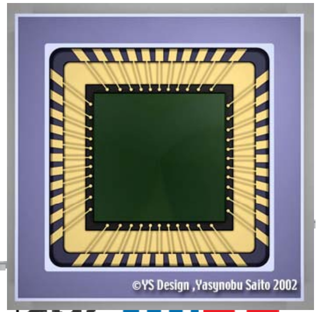




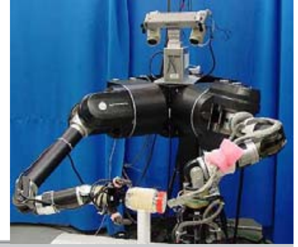
- CCD (charge coupled device)
  - Higher dynamic range
  - High uniformity
  - Lower noise

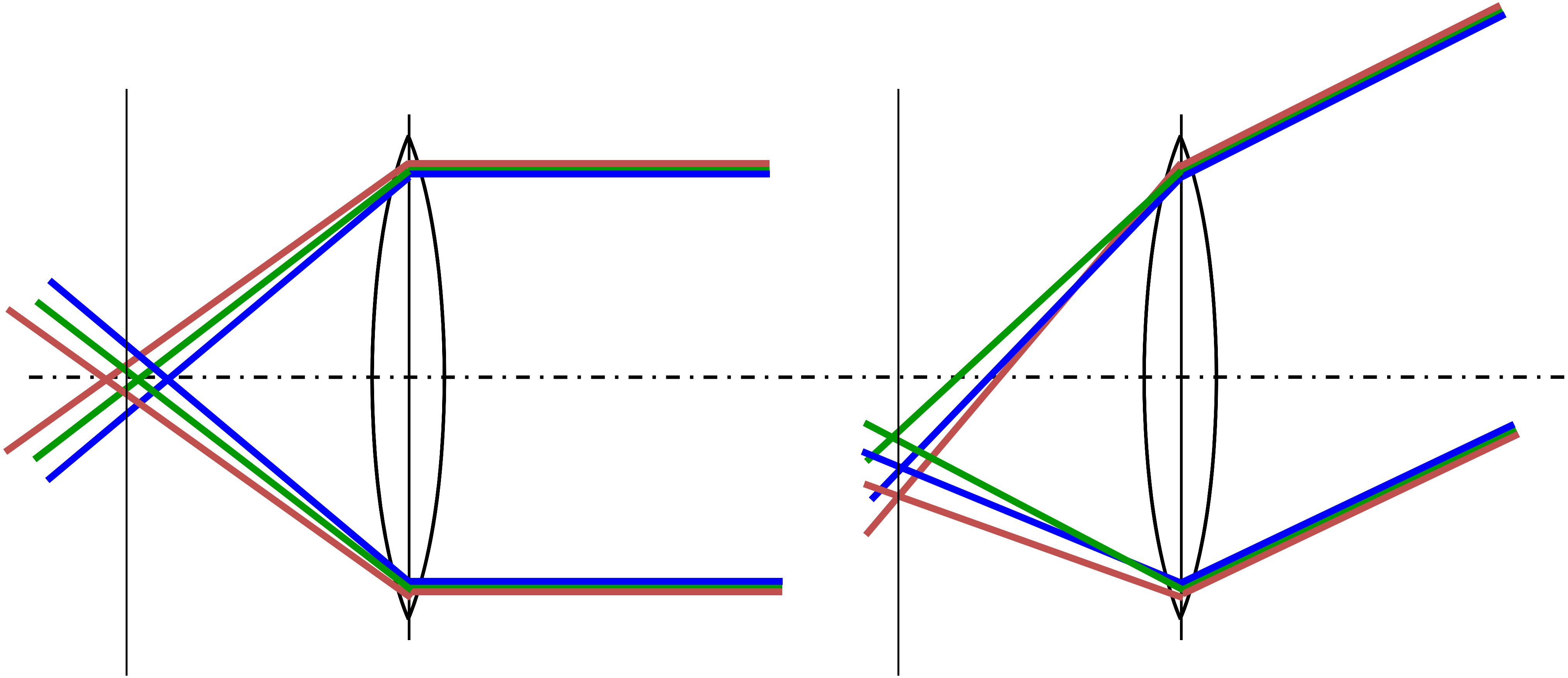


- CMOS (complementary metal Oxide semiconductor)
  - Lower voltage
  - Higher speed
  - Lower system complexity



Chromatic Aberration

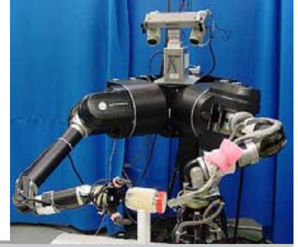




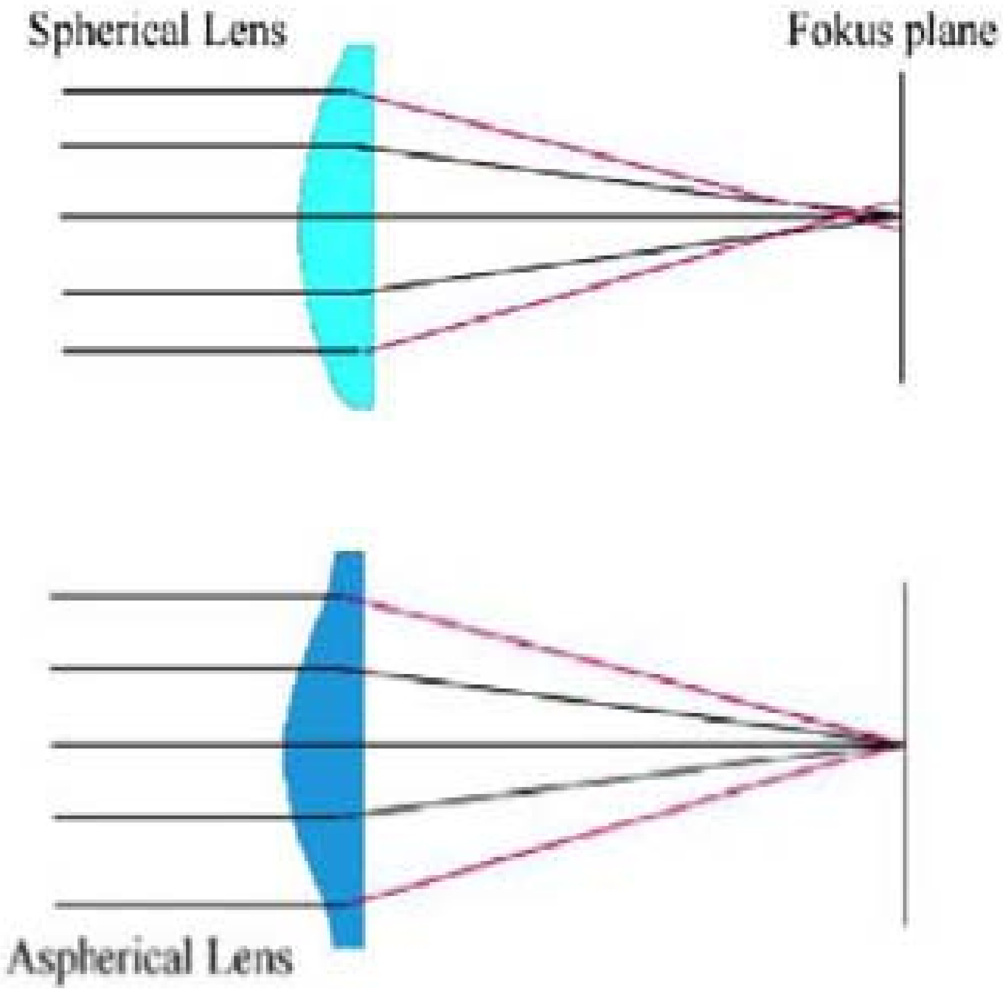
longitudinal chromatic aberration
(axial)

transverse chromatic aberration
(lateral)

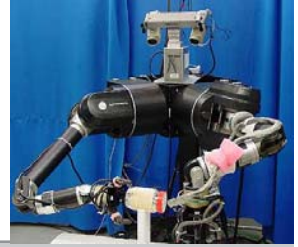
Spherical Aberration

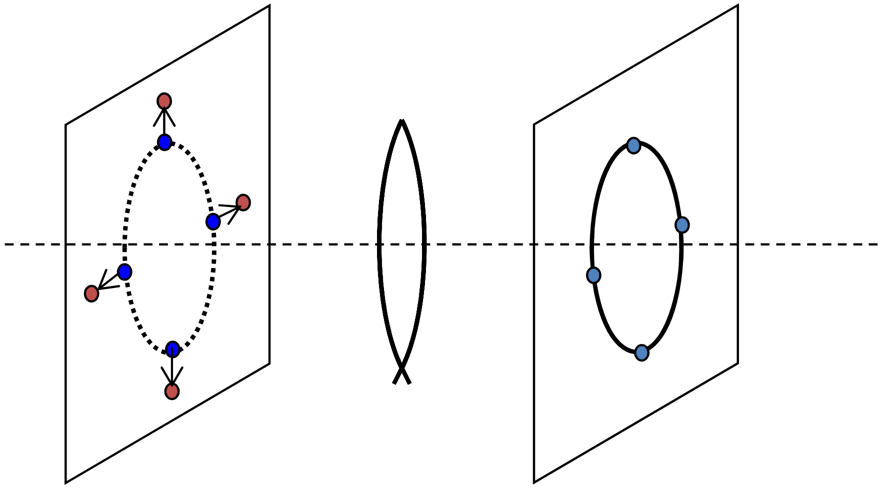


- Effect: sharp image superimposed by a blurred one
- Caused by spherical lens surfaces (manufacturing)
- Parallel rays are focused in one point only if they are close to the optical axis
- Can be avoided by using aspherical lenses with parabolic surfaces

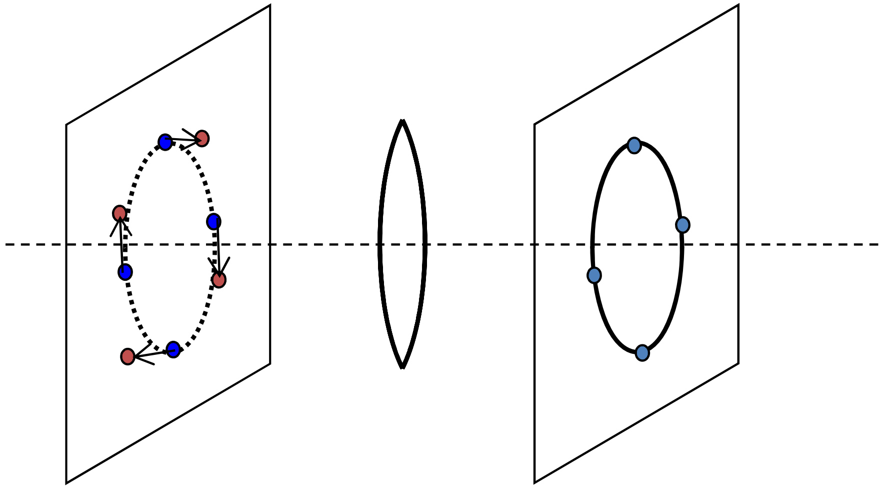


Geometric Lens Distortions





Radial distortion

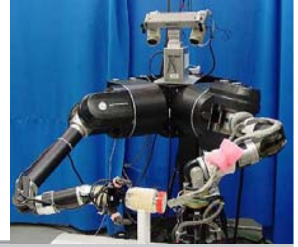


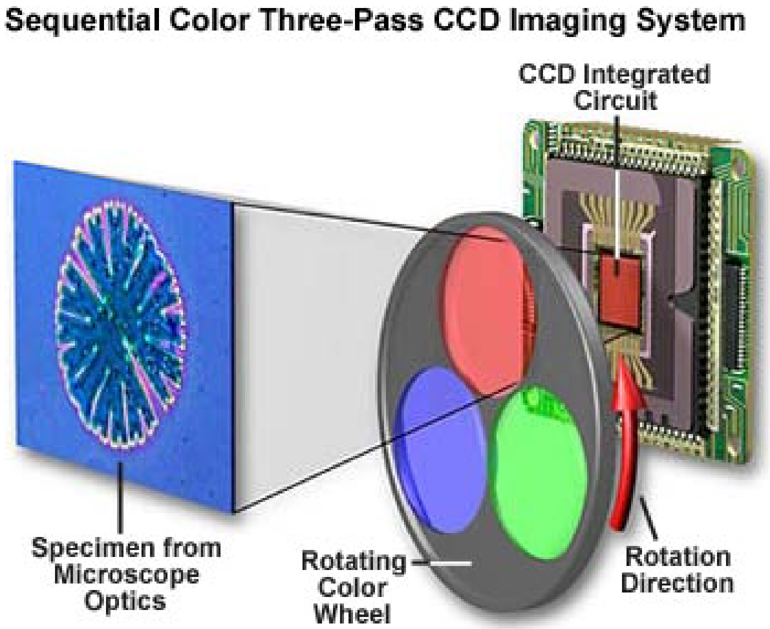
Tangential distortion

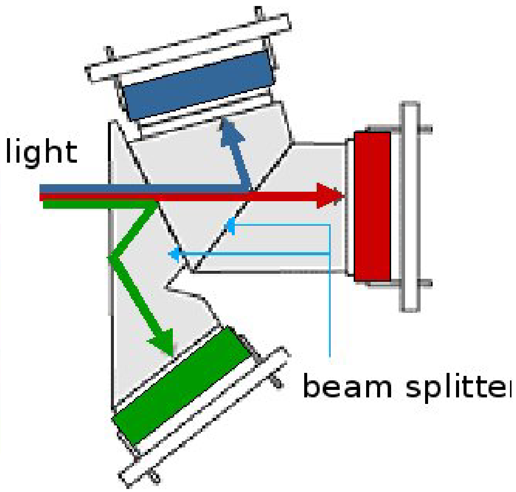
Photo by Helmut Dersch

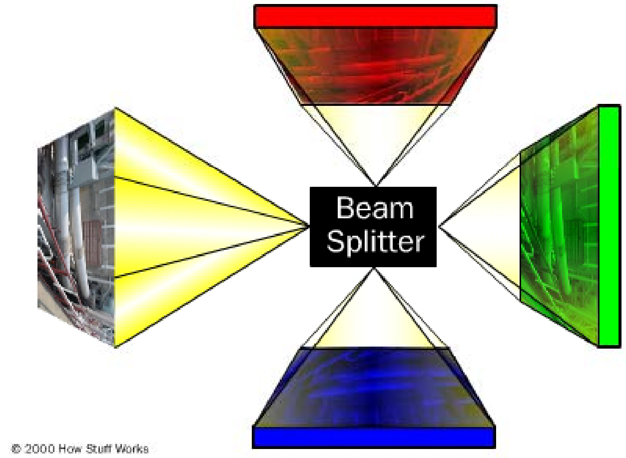
Both due to lens imperfection

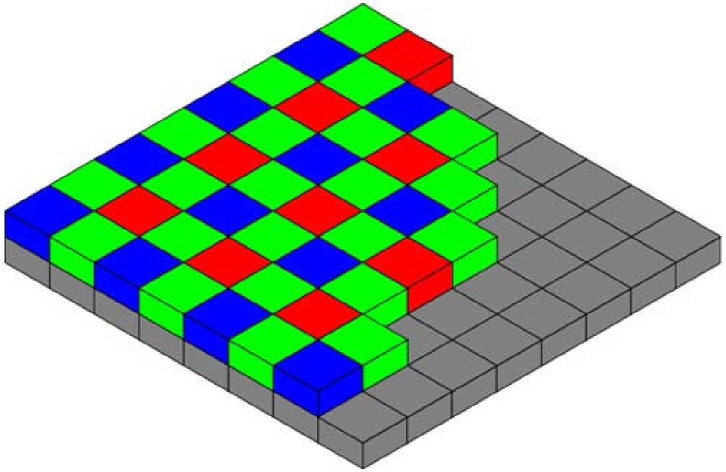
How CCDs Record Color

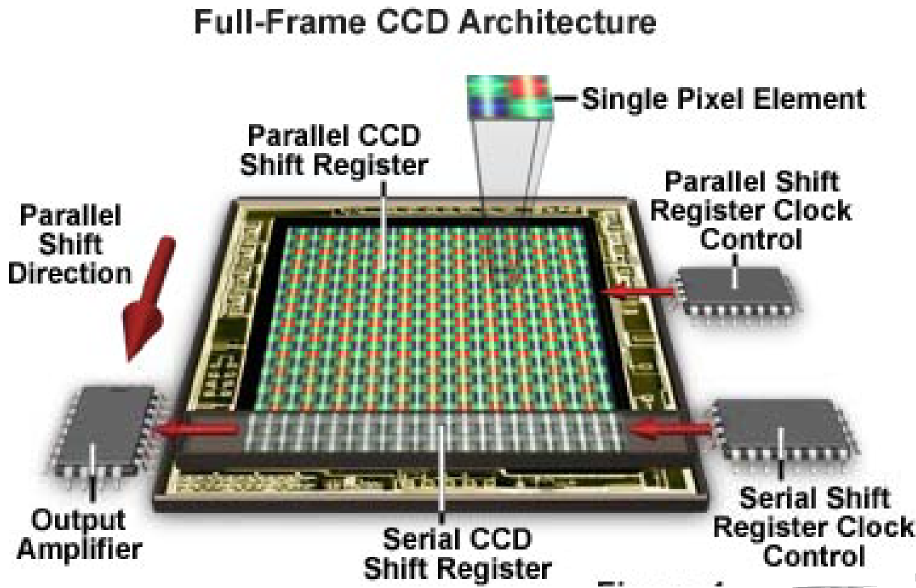




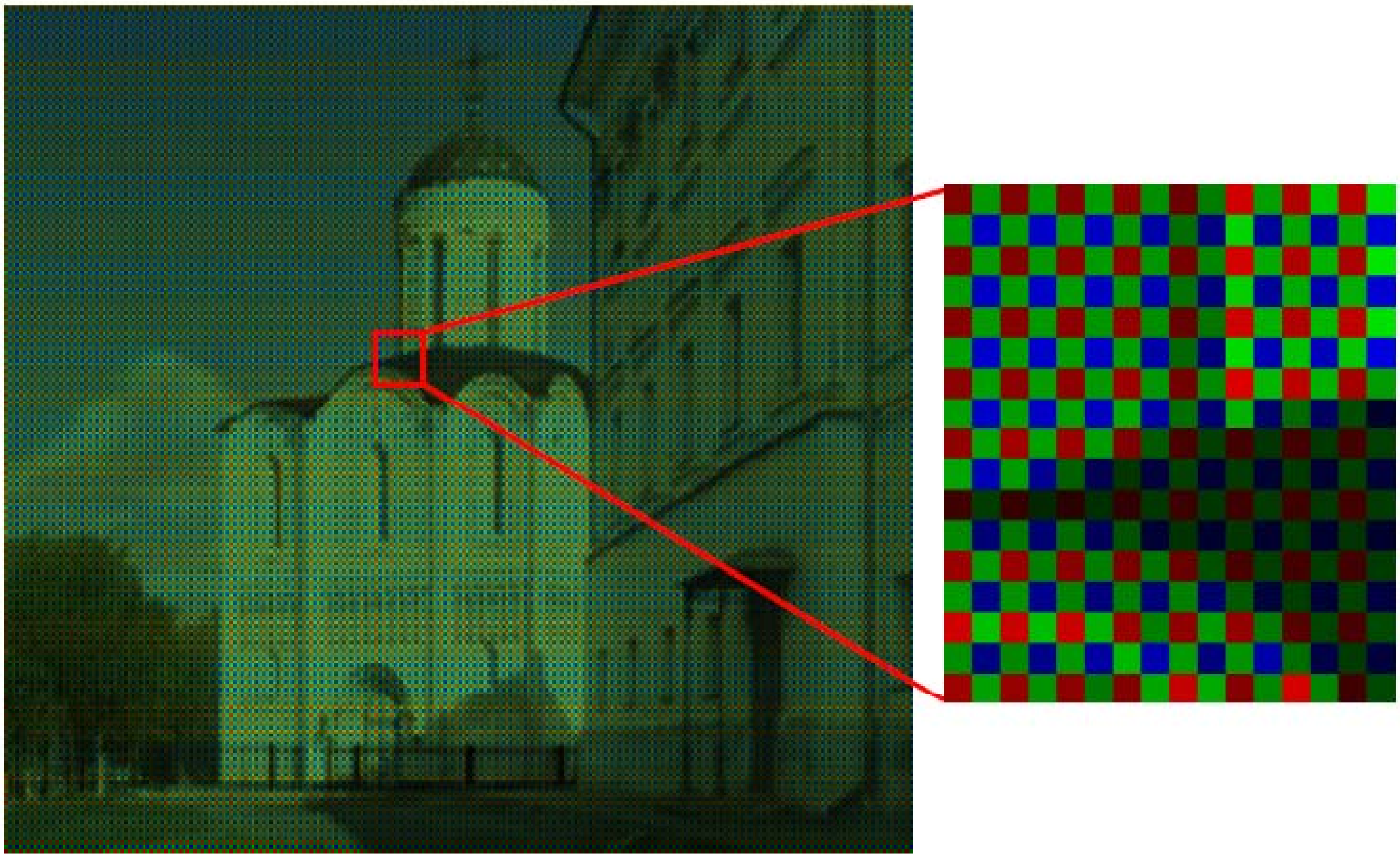




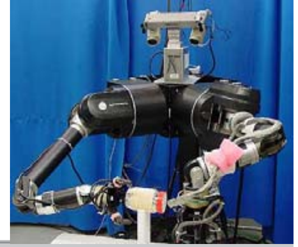




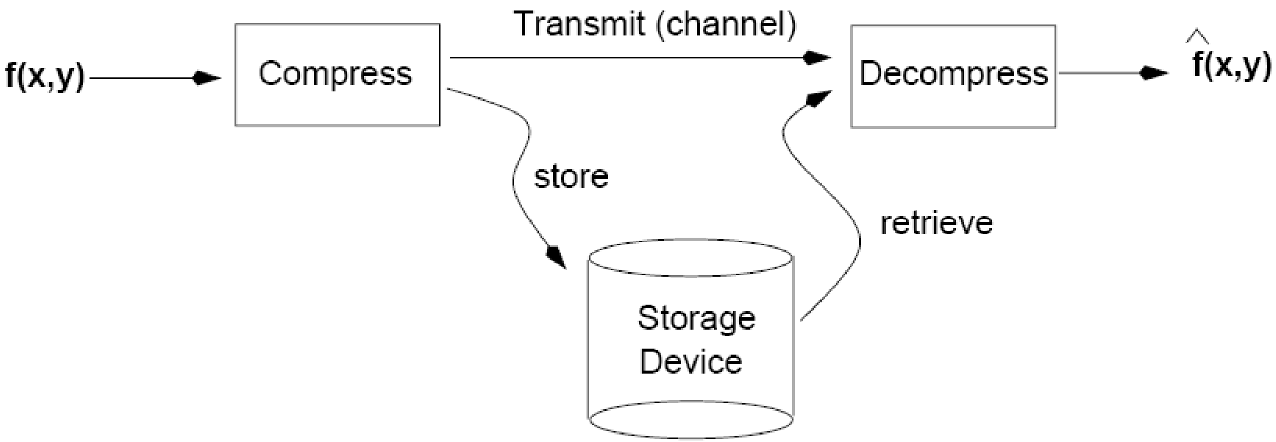
Bayer Filter



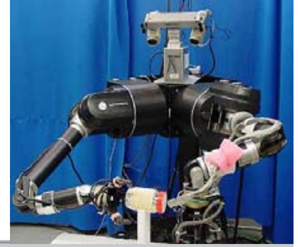
Goal of Image Compression



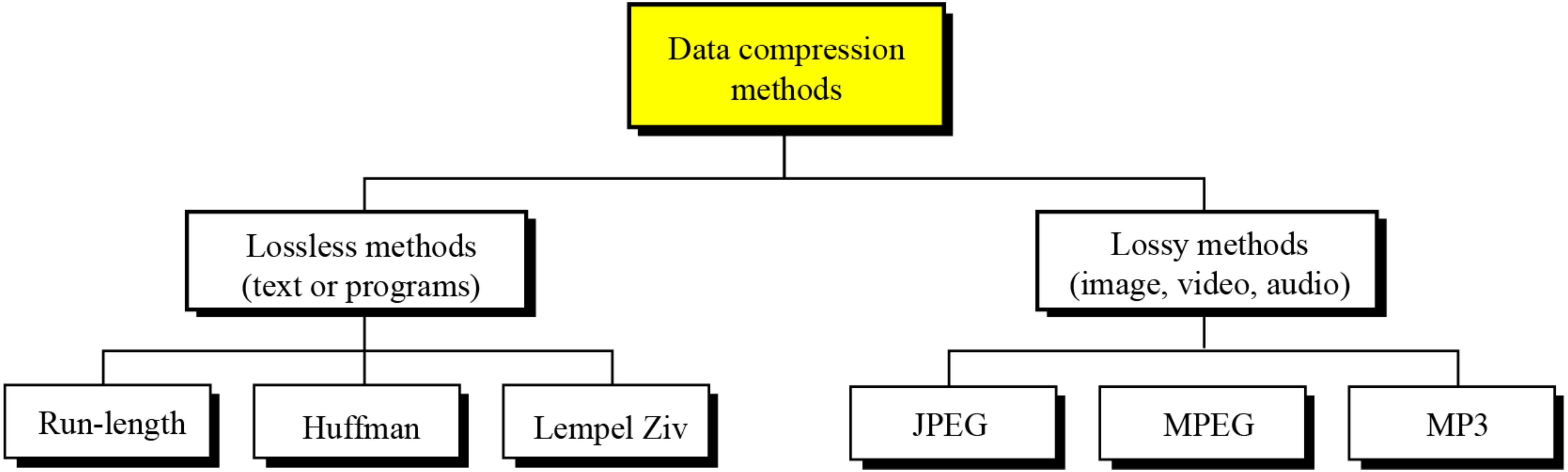
- Digital images require huge amounts of space for storage and large bandwidths for transmission.
  - A 640 x 480 color image requires close to 1MB of space.
- The goal of image compression is to reduce the amount of data required to represent a digital image.
  - Reduce storage requirements and increase transmission rates.

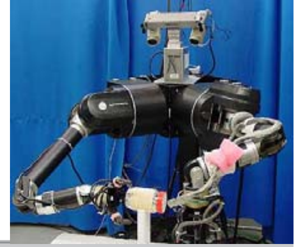


Data Compression



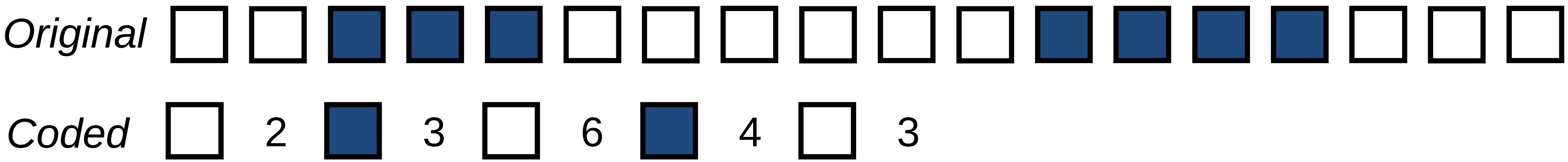
- Data compression implies sending or storing a smaller number of bits.
  - lossless and
  - lossy methods.
  - Trade-off: image quality **vs** compression ratio





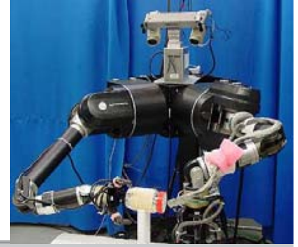
Run Length Encoding (RLE)

- Simplest method of compression
- Can be used to compress data made of any combination of symbols, does not need to know the frequency of occurrence of symbols
- Replace consecutive repeating occurrences of a symbol by one occurrence of the symbol followed by the number of occurrences

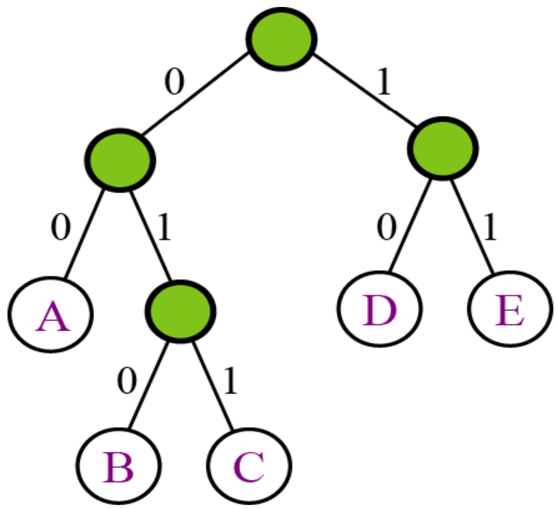


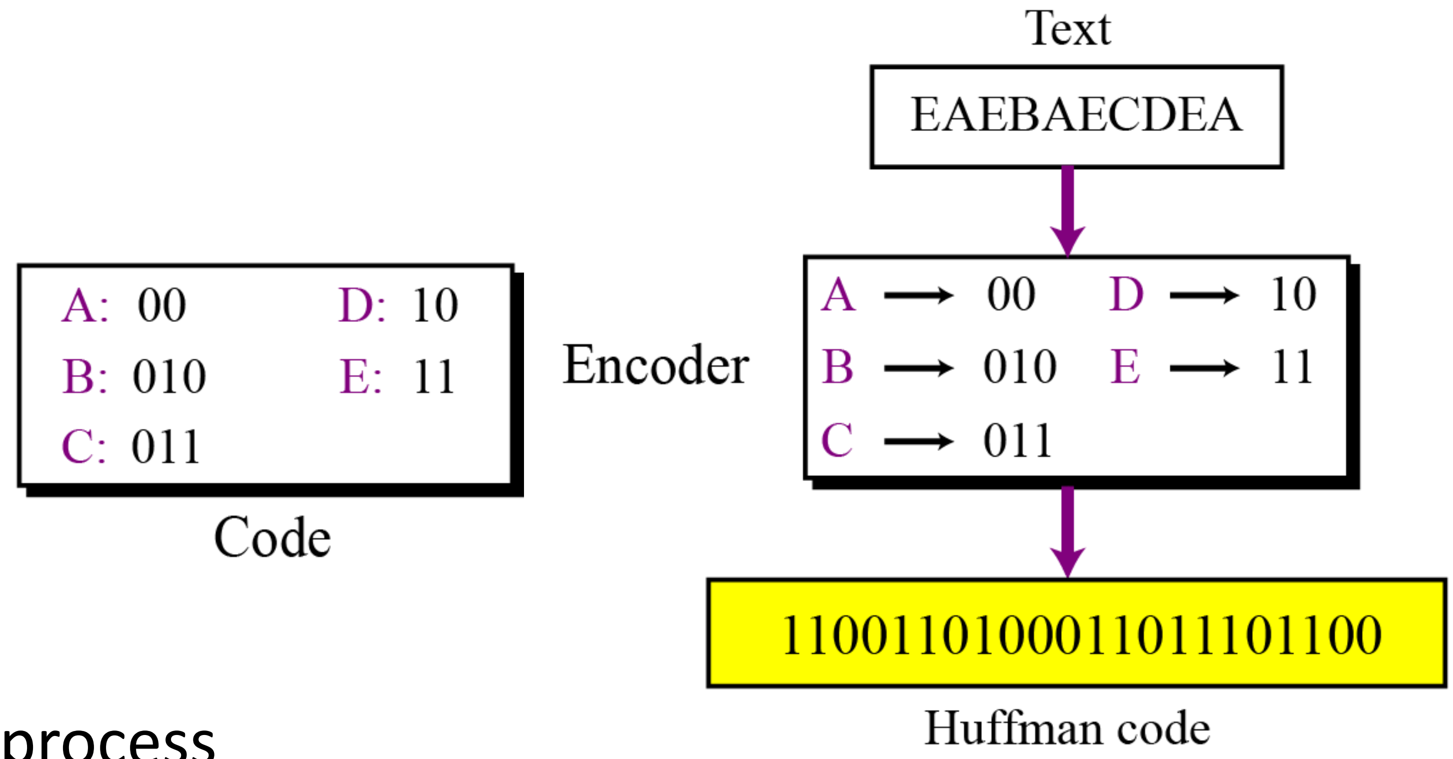
- Lossless compression!

Huffman Encoding



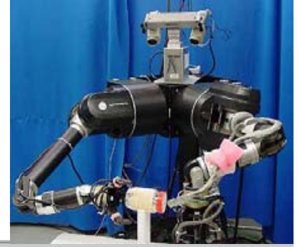
- Character code found by starting at the root and following the branches that lead to that character.
- The code itself is the bit value of each branch on the path, taken in sequence.



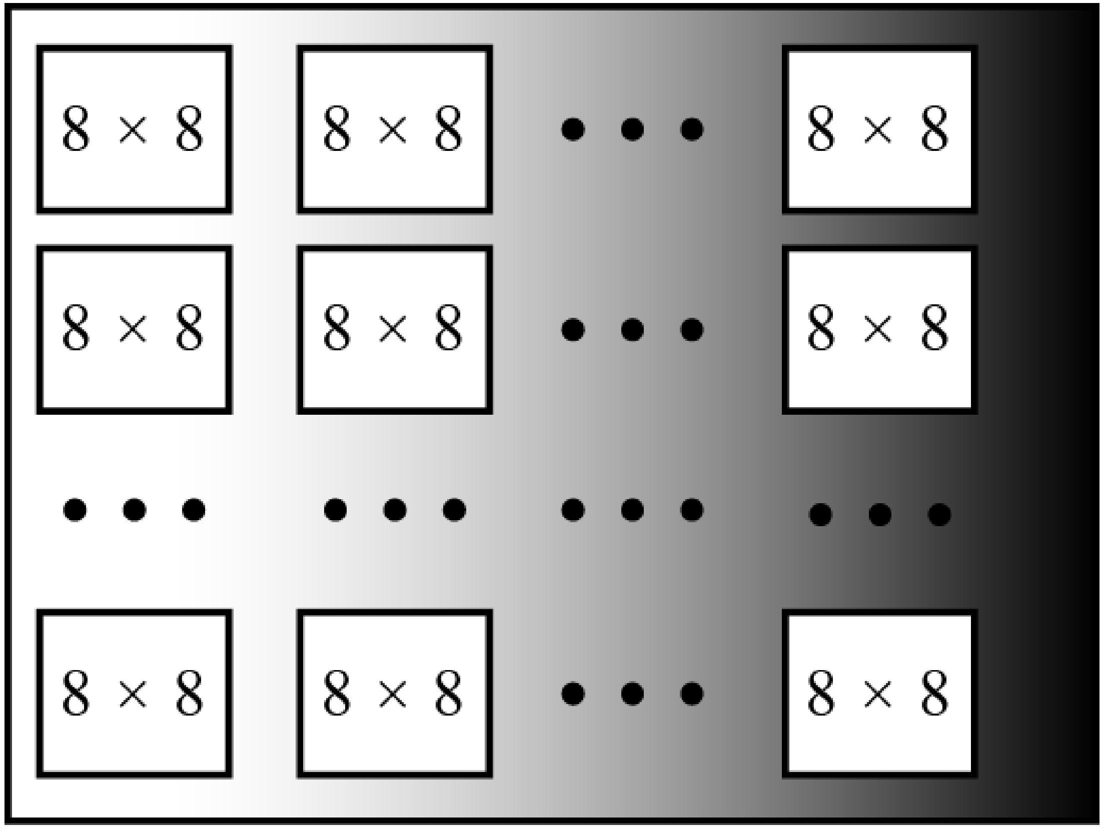


- Decoding: reverse process

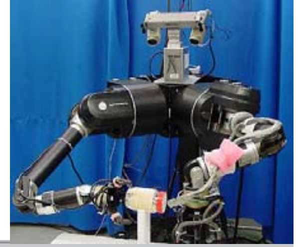
JPEG Compression



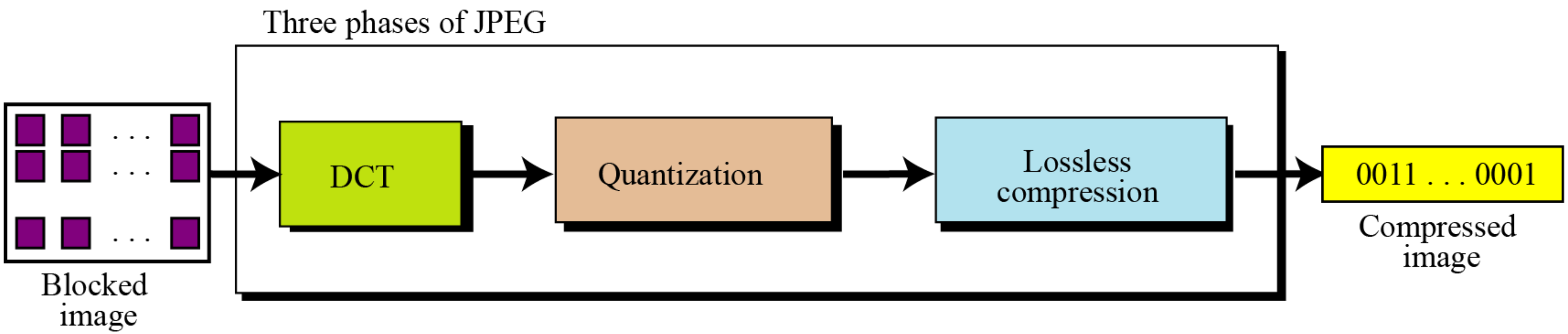
- Image is divided into blocks of 8 × 8 pixel blocks to decrease the number of calculations because the number of mathematical operations for each image is the square of the number of units.



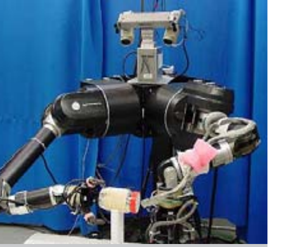
JPEG Compression

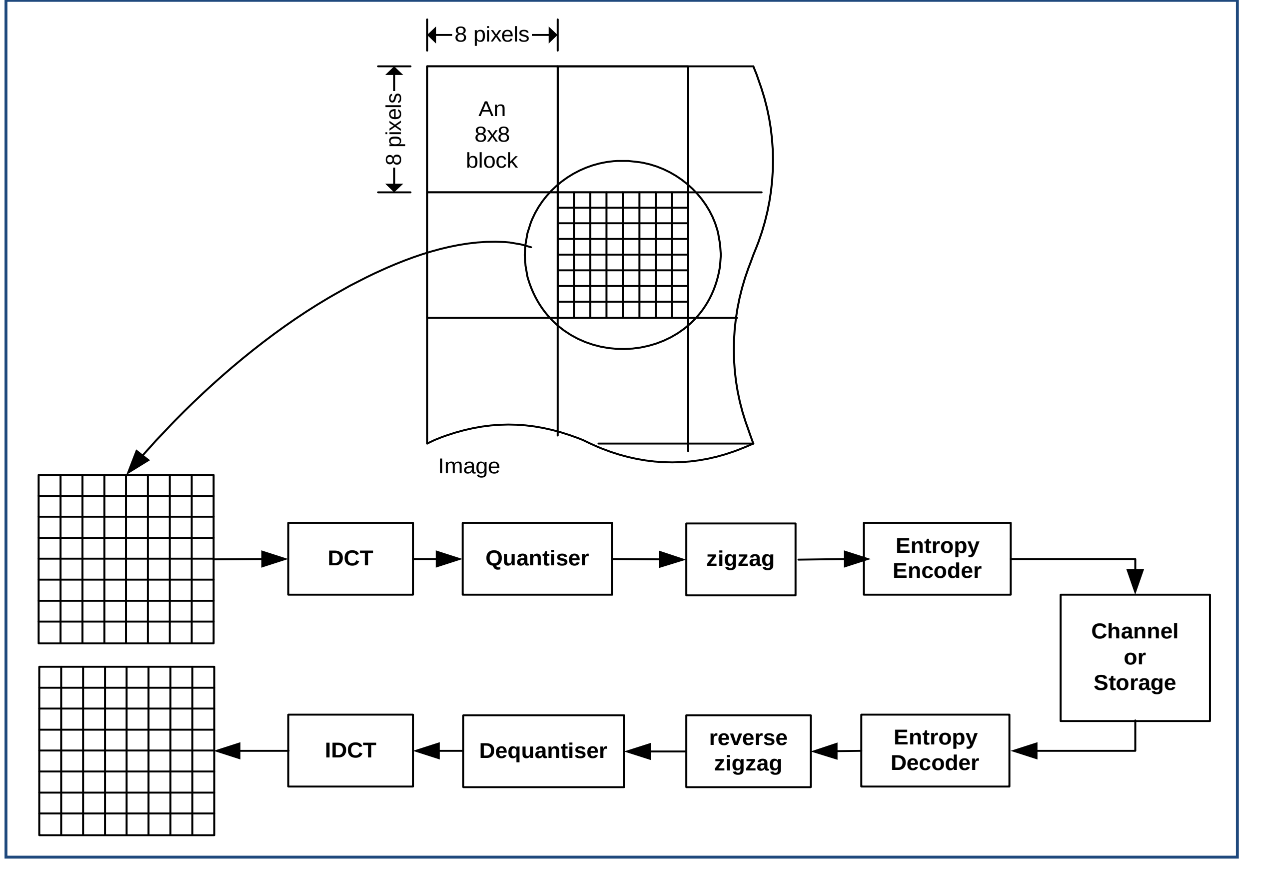


- Idea: change image into a linear (vector) set of numbers that reveals redundancies.
- Redundancies can be removed using one of the lossless compression methods



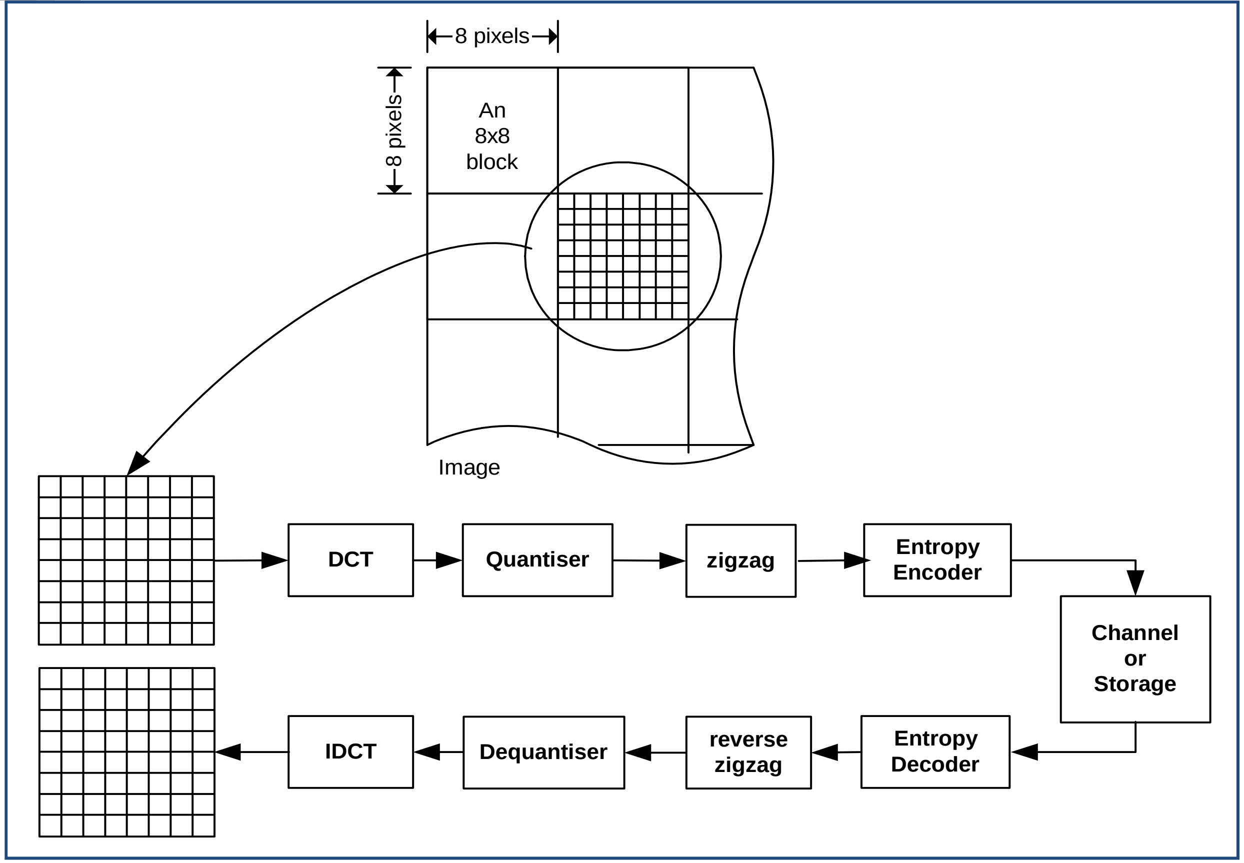
JPEG Compression





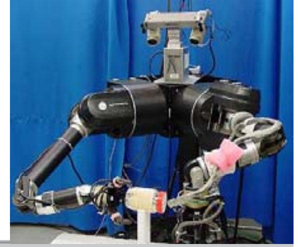
- To perform the JPEG coding, an image (in color or grey scales) is first subdivided into blocks of 8x8 pixels.
- The Discrete Cosine Transform (DCT) is then performed on each block.
- This generates 64 coefficients which are then quantized to reduce their magnitude.

JPEG Compression



- The coefficients are then reordered into a one-dimensional array in a zigzag manner before further entropy encoding.
- The compression is achieved in two stages; the first is during quantization and the second during the entropy coding process.
- JPEG decoding is the reverse process of coding.

Videocompression

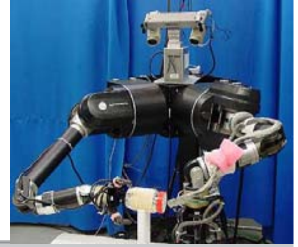


- Single Image
  - Size 720 x 576 px
  - Pixelresolution: 1 Byte/RGB Value

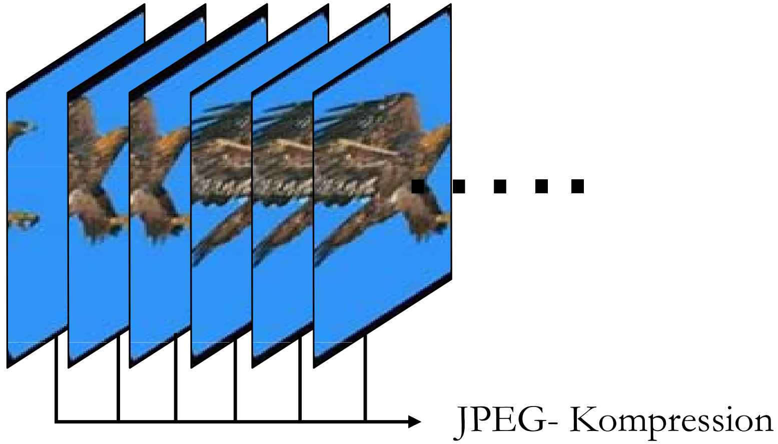
    → 720 x 576 x 3 (Byte) ~ 1.215 KB
- Image sequence
  - 25 fps

    → 720 x 576 x 25 x 3 (Byte) ~ 30.375 KB/s

M- JPEG



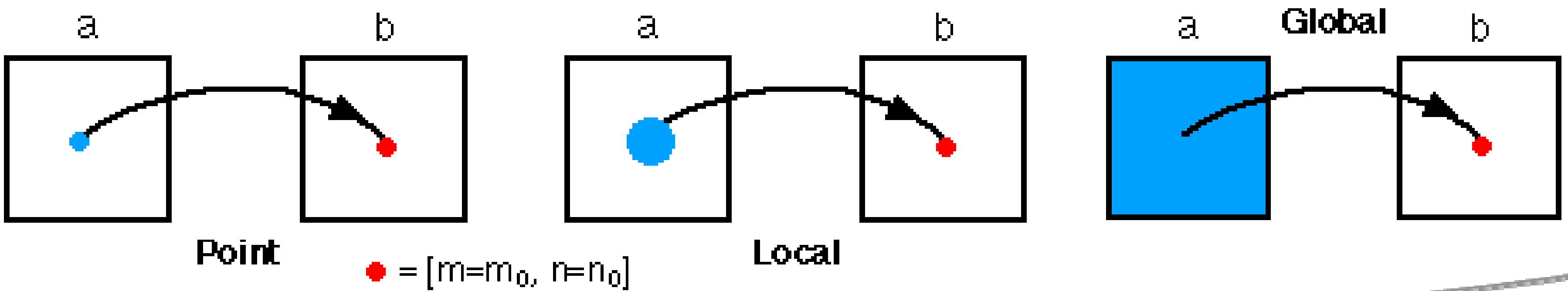
- Videosequenzes
- Single image compression using JPEG



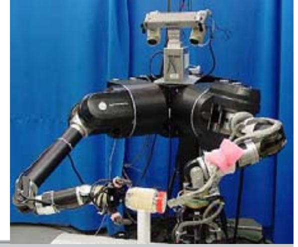
Benefits	Drawbacks
Constant Image Quality	Fluctuating bandwidth / frame rate
Fast computation	High memory requirements
Robust with respect to packet loss	No Audio

Types of Operations

- The operations that can be applied to digital images to transform an input image a[m,n] into an output image b[m,n]
  - **Point**: the output value at a specific coordinate is dependent only on the input value at that same coordinate.
  - **Local**: the output value at a specific coordinate is dependent on the input values in the neighborhood of that same coordinate
  - **Global**: the output value at a specific coordinate is dependent on all the values in the input image



Point Operations

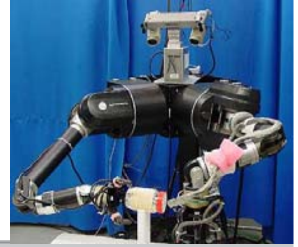


- Point Operations perform a mapping of the pixel values without changing the size, geometry, or local structure of the image
- Each new pixel value *I'(u,v)* depends on the previous value *I(u,v)* at the same position and on a mapping function *f()*
- The function *f()* is independent of the coordinates
- Such operation is called “homogeneous”

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

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

Threshold Operation



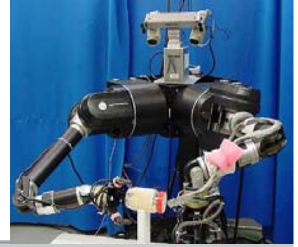
- Thresholding an image is a special type of quantization that separates the pixel values in two classes, depending on a given threshold value p_{th}
- The threshold function maps all the pixels to one of two fixed intensity values p_o, p_1

$$I'(u,v) \leftarrow f_{\text{th}}\big(I(u,v)\big) = \begin{cases} p_0 & \text{für} \;\; I(u,v) < p_{\text{th}} \\ p_1 & \text{für} \;\; I(u,v) \geq p_{\text{th}} \end{cases}$$

$$0 < p_{th} \leq p_{max}$$

- Example: binarization: $p_o=0$, $p_1=1$

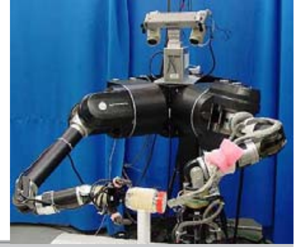
Histogram



- Assume that the digital image has *q* discrete gray levels and that n_k, $k = 0, \ldots, q\text{-}1$, is the number of pixel having intensity *k*. The histogram is given by:

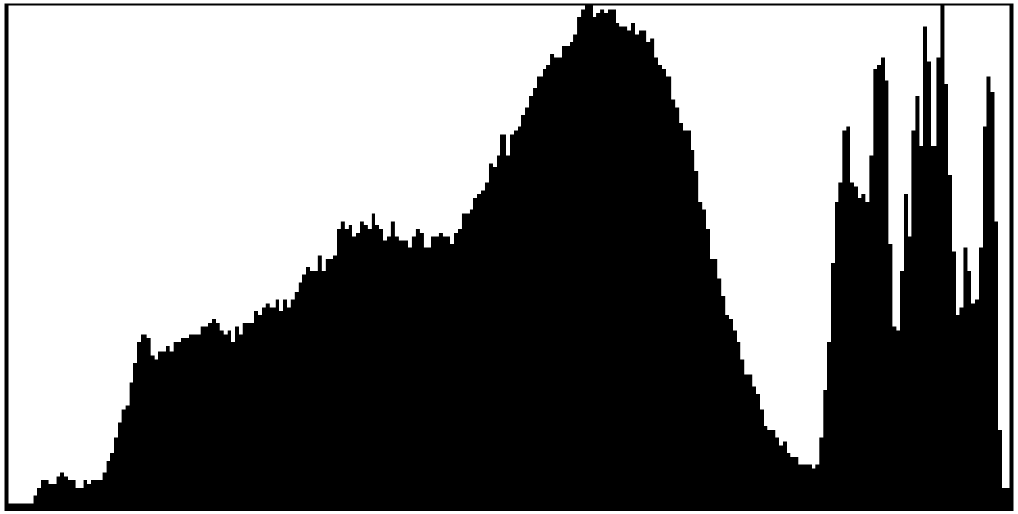
$$p(r_k) = \frac{h(r_k)}{n} = \frac{n_k}{n}$$

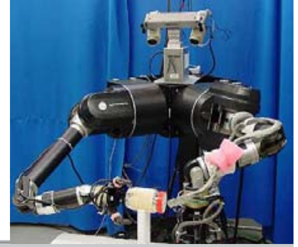
  where *p* is the normalized histogram function, *n* the total number of image pixels. n_k are the number of pixels in the bin assigned to pixels with intensity level *k*.
- It gives a measure of how likely is for a pixel to have a certain intensity. That is, it gives the probability of occurrence the intensity.
- The sum of the normalized histogram function over the range of all intensities is 1.



Histogram

- The histogram function can be plotted graphically. The image histogram carries important information about the image content.



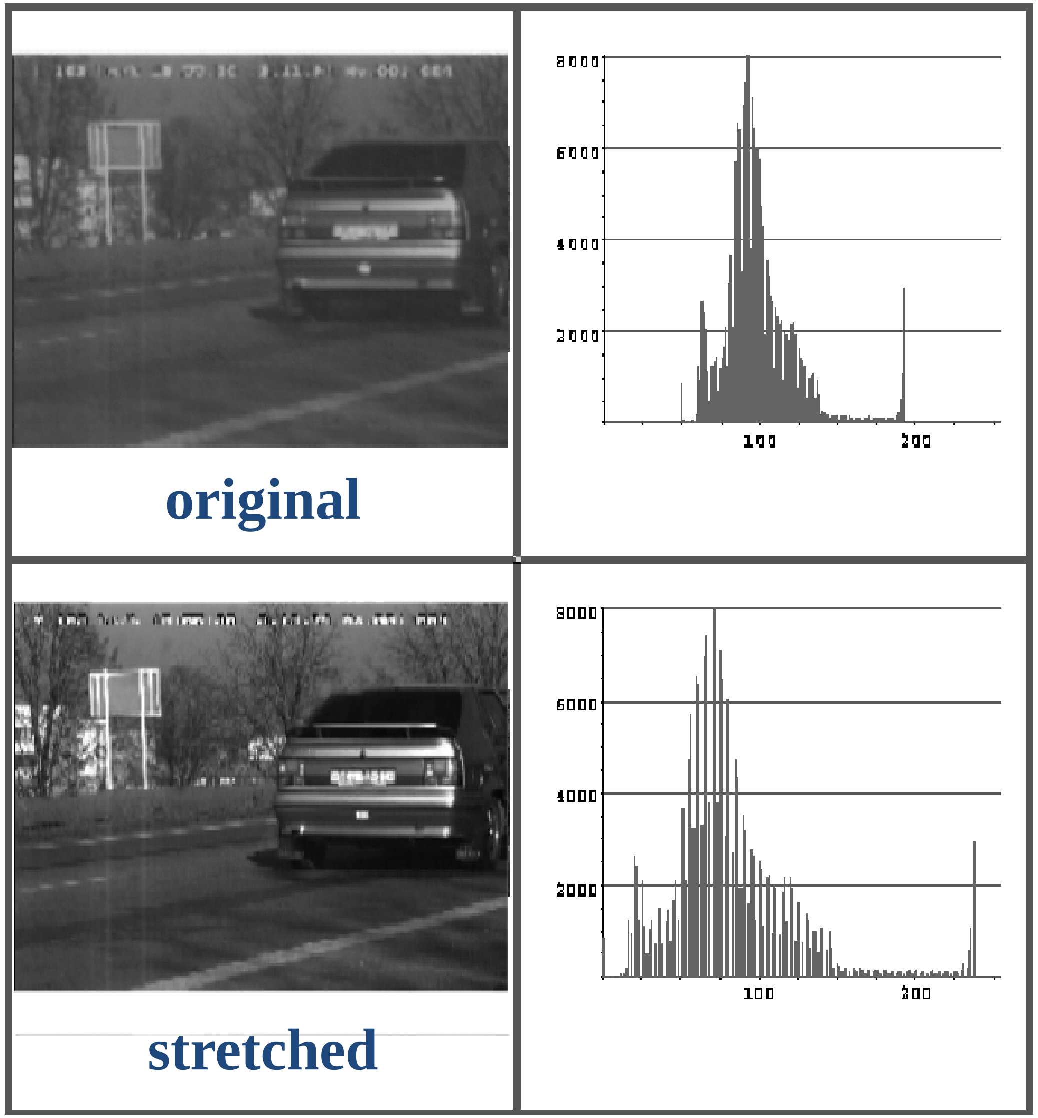


Histogram Normalization

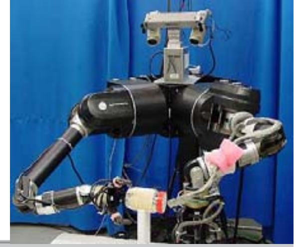
- Goal: utilization of the complete gray level range

- => linear spreading of gray levels to the complete gray level range

$$I'(u,v) = q \cdot \frac{I(u,v) - q_{min}}{q_{max} - q_{min}}$$



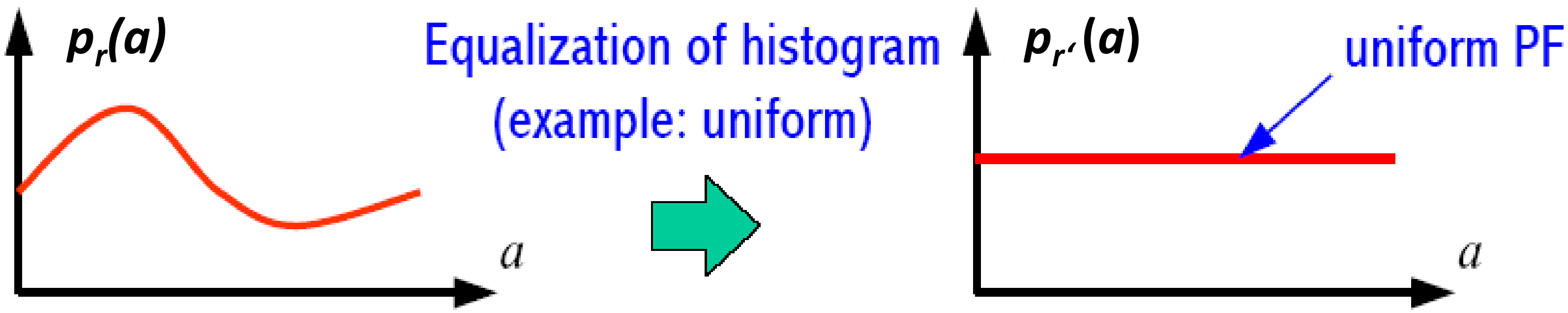
Histogram Equalization



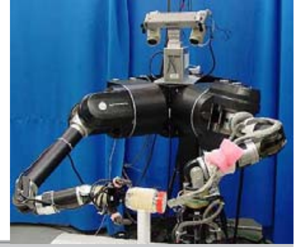
- Let $r(x,y)$ be a gray-level image whose minimum intensity value is r_{min} and maximum intensity value r_{max}. The **dynamic range** of the image Δr is: $\Delta r = r_{max} - r_{min}$.
- The **Probability Function** (PF) of the image $r(x,y)$ is $p(r = a) = p_r(a)$. With a probability of $p_r(a)$ the image takes a gray-level equal to the value of a.
- **Histogram equalization** means that we need to find a intensity level transforming function $T(a)$ that for the transformed image $r'(x,y)$ can be computed as

$$r'(x,y) = T(r(x,y))$$

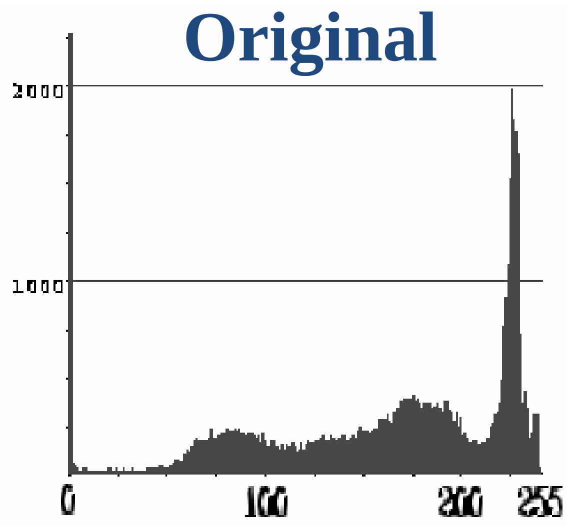
- $T(a)$ is chosen in way that the probability function $p_{r'}(a)$ of the transformed image $r'(x,y)$ has a predefined shape.

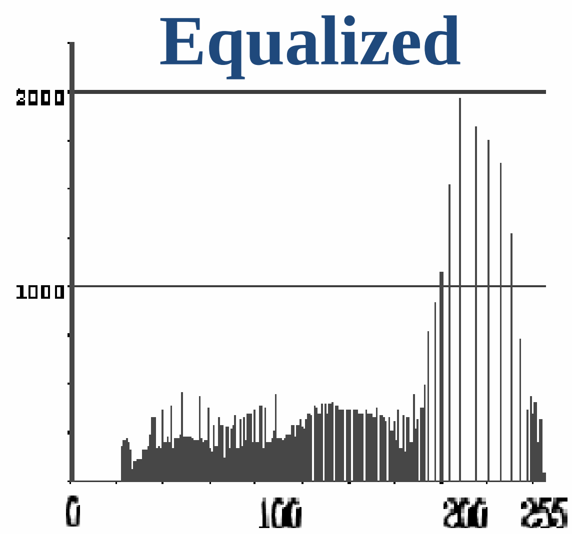


Histogram Equalization Example



- Goal: Equal Distribution of gray levels over the complete gray level range
- => Contrast is enhanced at maxima and weakened at minima

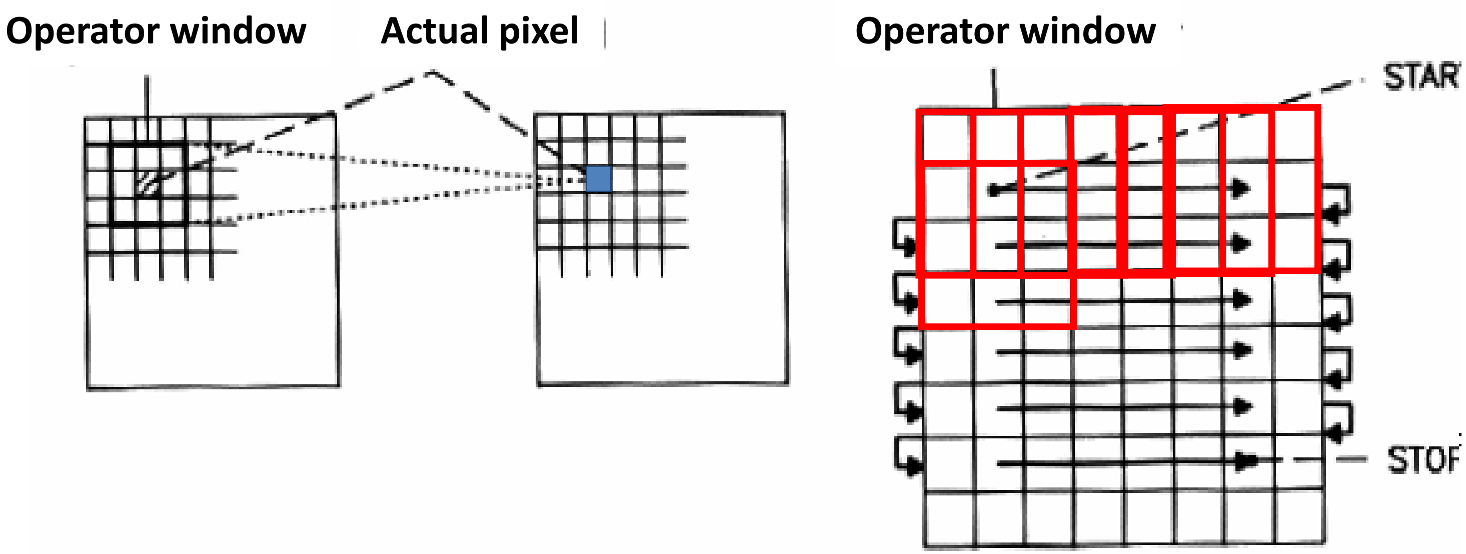




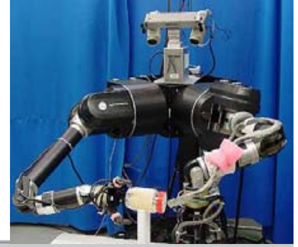
Local-Operators

The Gray level of the resulting pixel is dependent on several pixel in the original image:

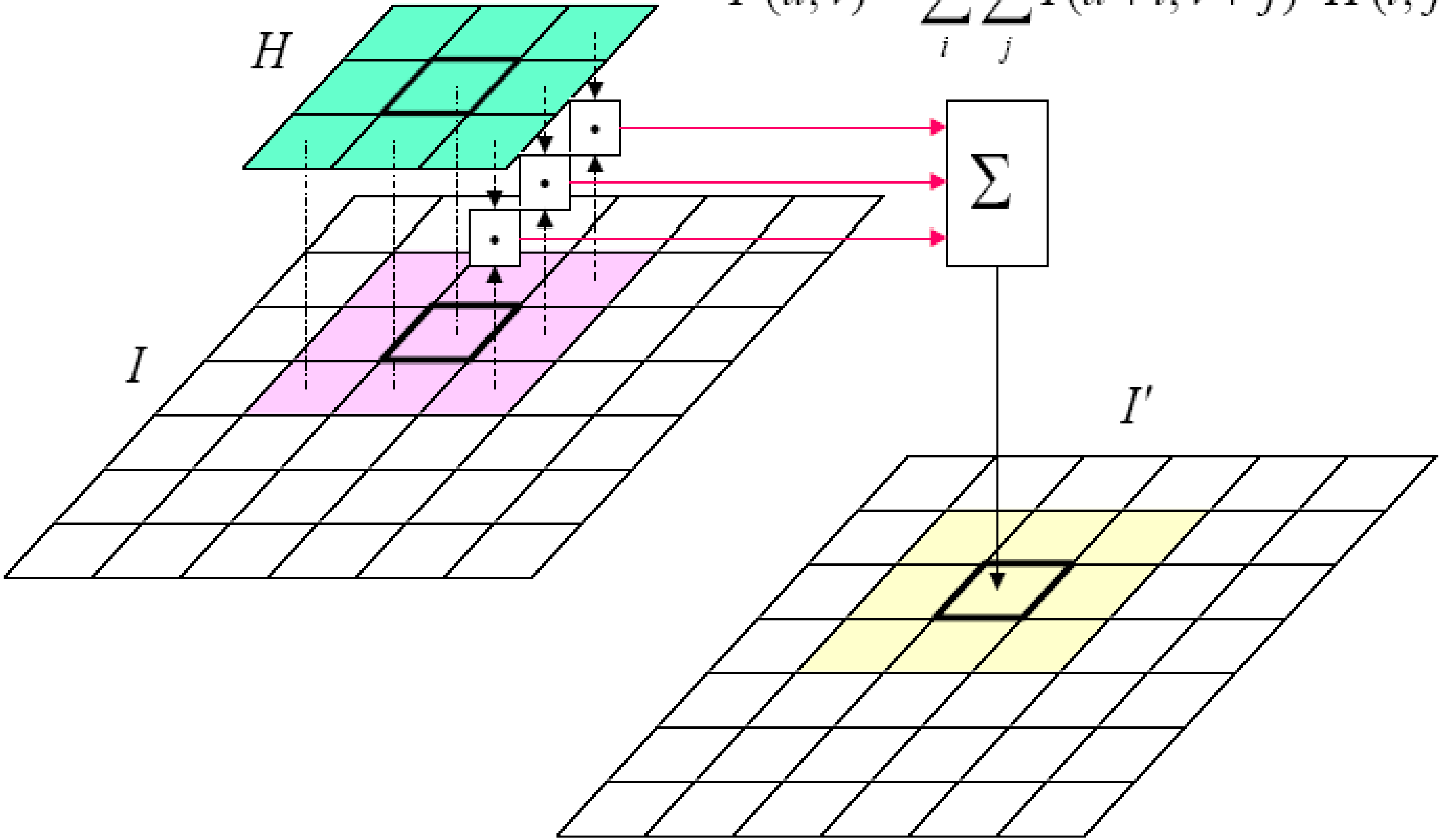
- An operator window is placed around an actual pixel
- Computation of the resulting pixel by combination of gray levels of the actual pixel and its neighbors and saving it at the position of the actual pixel at its position in the result image
- Computation of the complete image (by displacement of the operator window by one pixel)



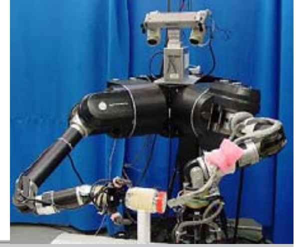
Linear 2-D Filter

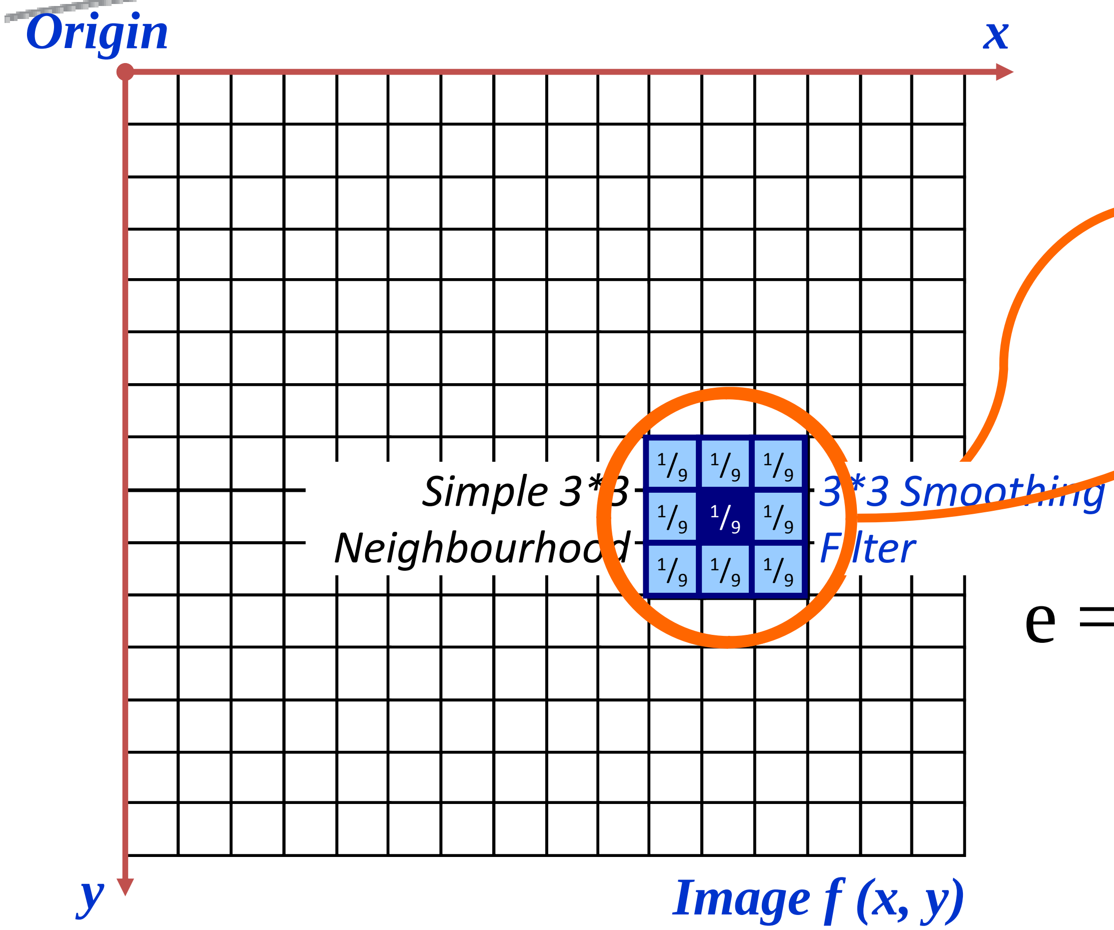


$$I'(u,v) = \sum_i \sum_j I(u+i, v+j) \cdot H(i,j)$$



Smoothing Spatial Filtering





Original Image Pixels

104	100	108
99	106	98
95	90	85

*

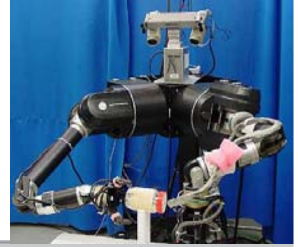
Filter

$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$

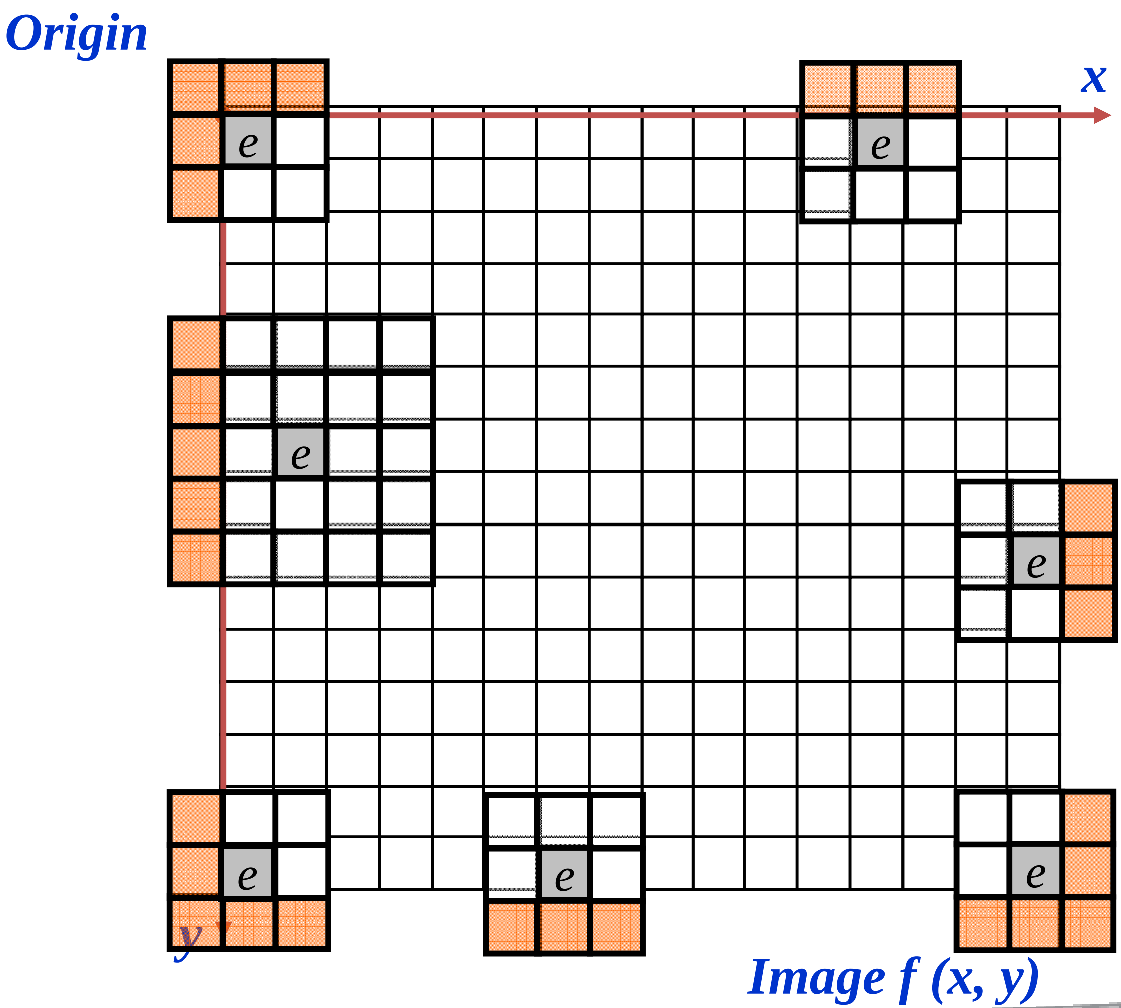
$$e = {}^1/_9 * 106 + {}^1/_9 * 104 + {}^1/_9 * 100 + {}^1/_9 * 108 + {}^1/_9 * 99 + {}^1/_9 * 98 + {}^1/_9 * 95 + {}^1/_9 * 90 + {}^1/_9 * 85 = 98.3333$$

The above is repeated for every pixel in the original image to generate the smoothed image

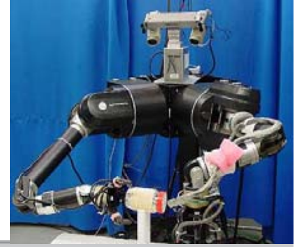
Strange Things Happen At The Edges!



At the edges of an image we are missing pixels to form a neighbourhood



Two-dimensional Convolution



$$I'(u,v) = \sum_i \sum_j I(u+i, v+j) \cdot H(-i,-j)$$

Function I

Function H

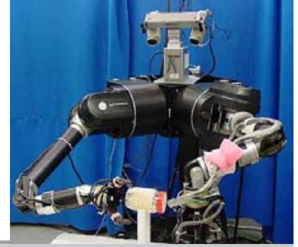
$$I' = I * H$$

Linerar Convolution

Properties of linear Filters

- Commutative:
  - $I * H = H * I$
- Linear:
  - $(a \cdot I) * H = I * (a \cdot H) = a \cdot (I * H)$
  - $(I_1 + I_2) * H = (I_1 * H) + (I_2 * H)$
- Associative:
  - $A * (B * C) = (A * B) * C$
  - Separabel: $H = H_1 * H_2 * . . . * H_n$

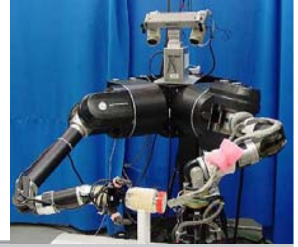
Image Filters



- Filter kernels are scaled in order to guarantee that the brightness does not change generally. The scaling factor is computed by the sum of the coefficients of the kernel matrix.
- Example:

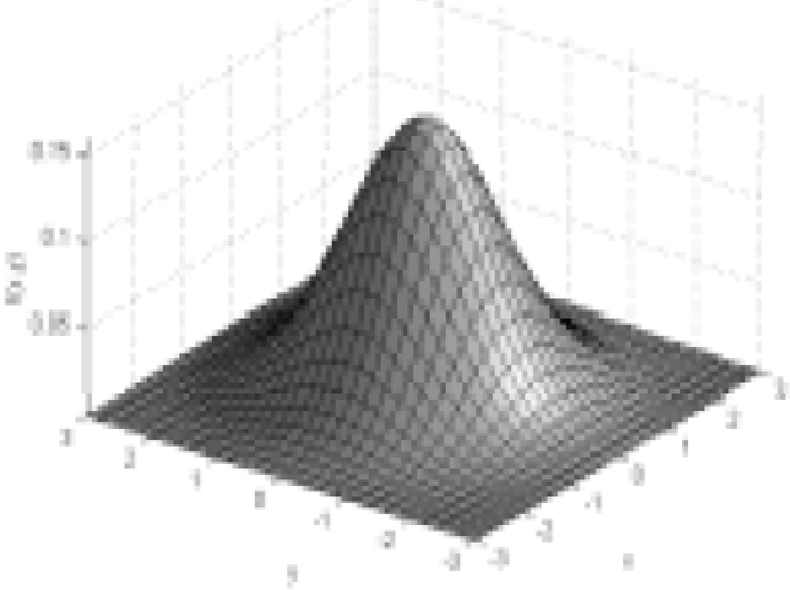
□ Intensity of a pixel is the sum of all of its 8 neighbors □ On a white image (all pixel = 1) new value 9!	$K = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$
□ Intensity of a pixel is the average of all of its 8 neighbors □ On a white image (all pixel = 1) new value 1!	$K = \frac{1}{9} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

Gaussian Filter



- Filter kernel can be approximated by convolution of two one-dimensional binomial distributions.

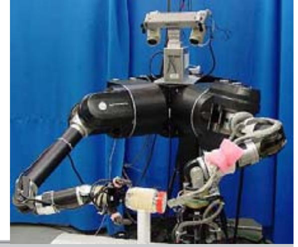
Approximation to Gauss‘ bell-shape curve (2D)



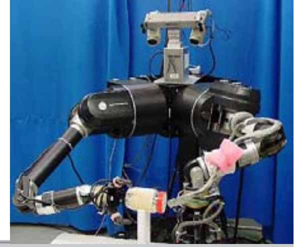
Example: 3x3 Gaussian Kernel

$$K = \frac{1}{16} \cdot \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Gaussian Filter



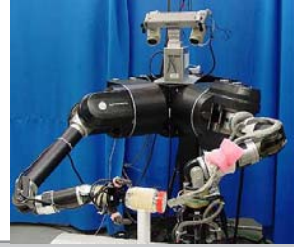
- Gaussians are used because:
  - Smooth (infinitely differentiable)
  - Decay to zero rapidly
  - Simple analytic formula
  - Separable: multidimensional Gaussian = product of Gaussians in each dimension
  - Convolution of 2 Gaussians = Gaussian
  - Limit of applying multiple filters is Gaussian (Central limit theorem)



Non-linear Filters

- Linear filters may have disadvantages: Linear smoothing filter suppresses noise but blurs the image at the same time.
- Non-linear filters (rank value filters) do not have this disadvantage. Therefore they are usually applied for noise removal.

Median Filter



The Median filter is a non-linear filter. The kernel window selects a pixel set *S* around the center coordinates (x_c, y_c) of the original image. The new value $C'(x_c, y_c)$ is set to the median value of the pixel set *S*. That is $C'(x_c, y_c) = median(S)$.

Image

0	0	1	2	3	2	5	3	2	2	1	4
7	4	9	0	7	6	2	3	4	6	7	8
9	8	6	5	4	3	9	7	9	6	3	3
5	6	6	5	3	3	8	9	3	3	4	7
8	5	4	3	3	2	3	3	4	5	6	7
8	4	6	9	3	0	6	4	3	6	7	9
1	2	3	2	1	1	0	2	2	3	4	2
4	5	6	7	5	4	3	2	4	6	5	3

values from window

7	6	2	3	4	4	3	9	7	9	3	3	8	9	3	3	2	3	3	4	3	0	6	4	3

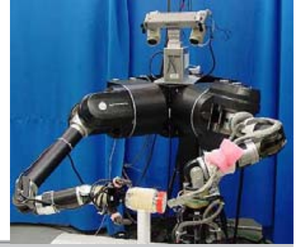
sorted values

0	2	2	3	3	3	3	3	3	3	3	3	3	4	4	4	4	6	6	7	7	8	9	9	9

minimum „middle“ maximum

→ Median → new center value

Median Filter: Outliers are eliminated



0	0	0	0	0
0	0	0	0	0
0	0	0	1	0
0	0	0	0	0
0	0	0	0	0

3x3 Smooth →

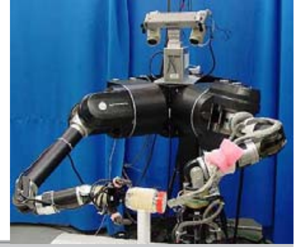
0	0	0	0	0
0	0	1/9	1/9	1/9
0	0	1/9	1/9	1/9
0	0	1/9	1/9	1/9
0	0	0	0	0

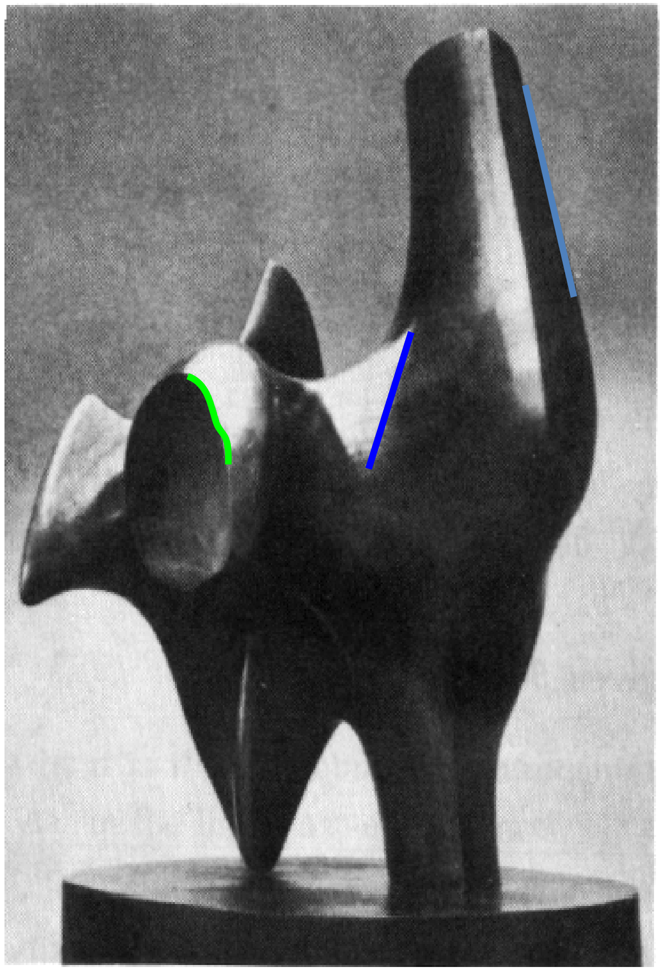
3x3 Median →

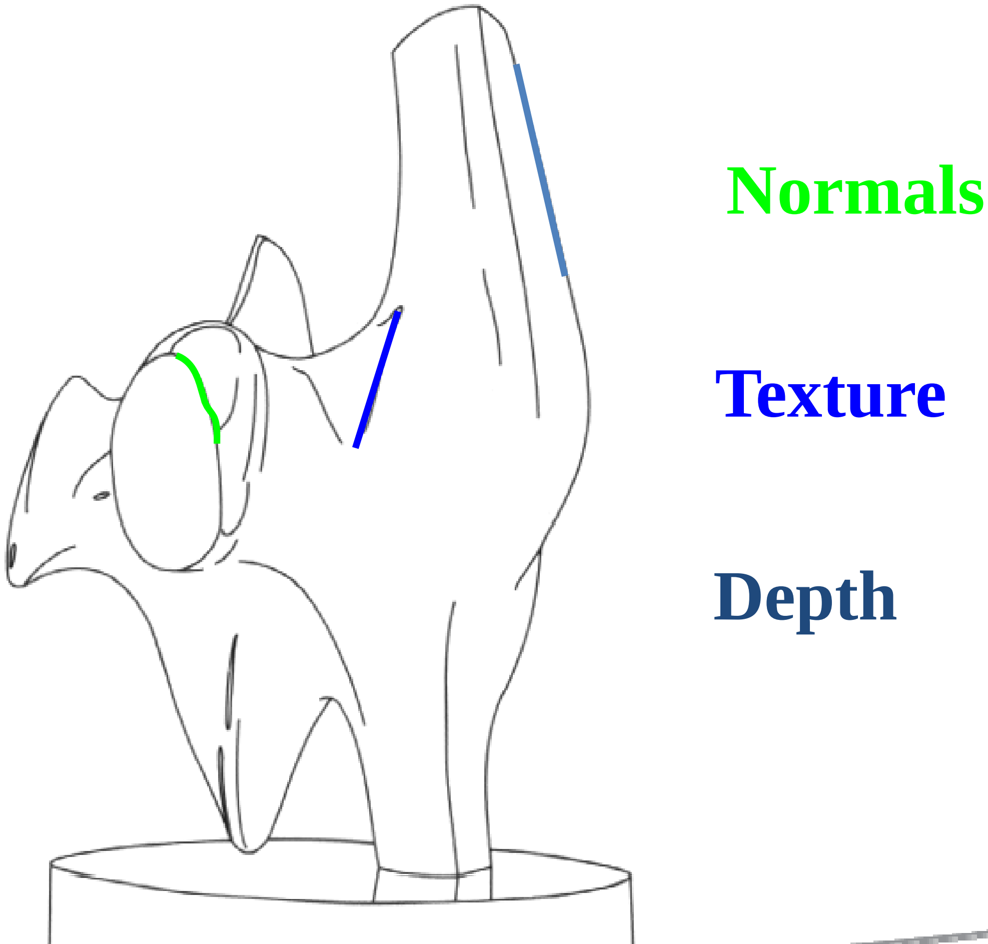
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Edge Detection

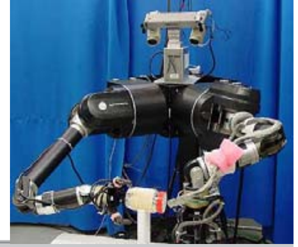
- Edges describe colloquially the edge of a surface or a significant change in orientation of the surface normals



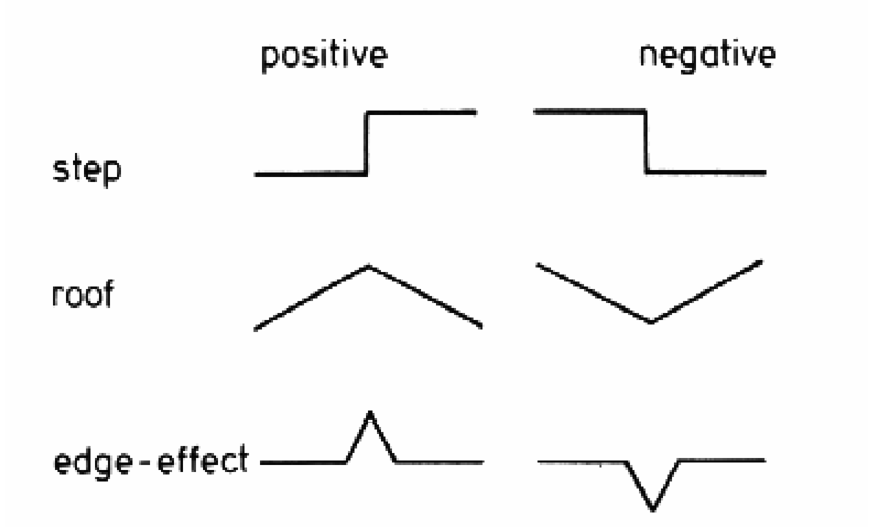




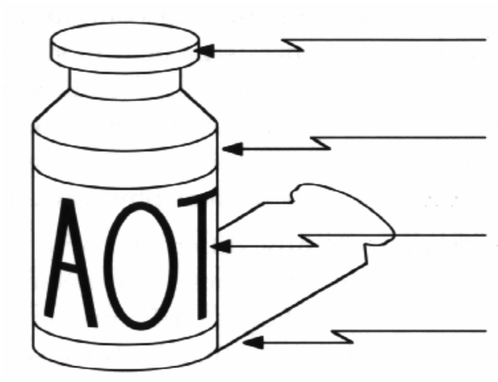
Types of Edges



- Step and "roof" edges
- Direction (positive or negative)
- Difference between gray values along an edge: **Contrast**

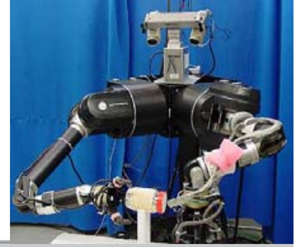


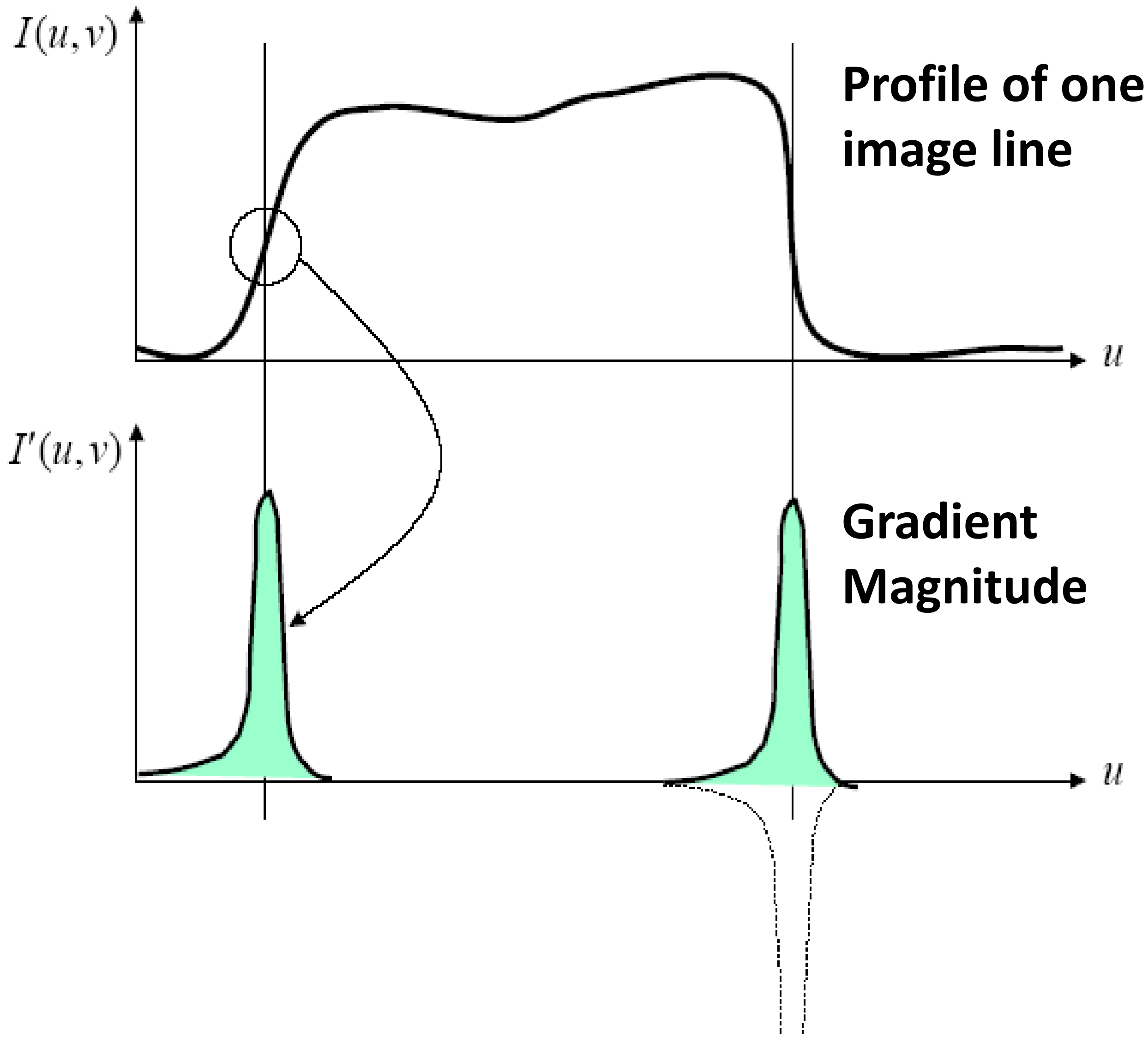
- **Discontinuities** of the brightness represent:



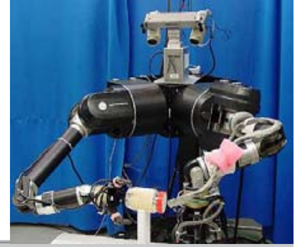
Discontinuities of surface **normal**
Discontinuities of the **depth**
Discontinuities of **texture**
Discontinuities of **illumination**

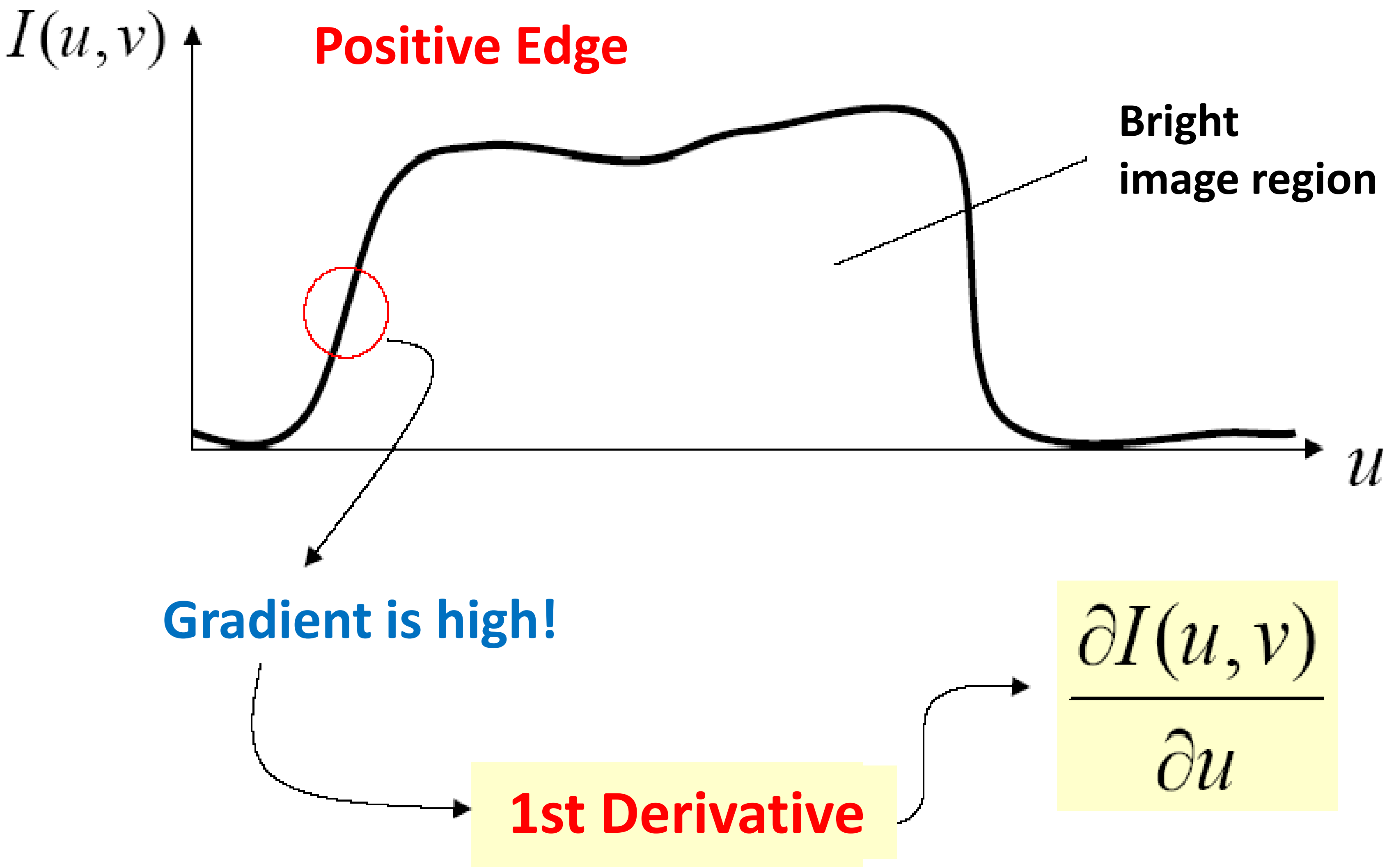
Edge-derivative





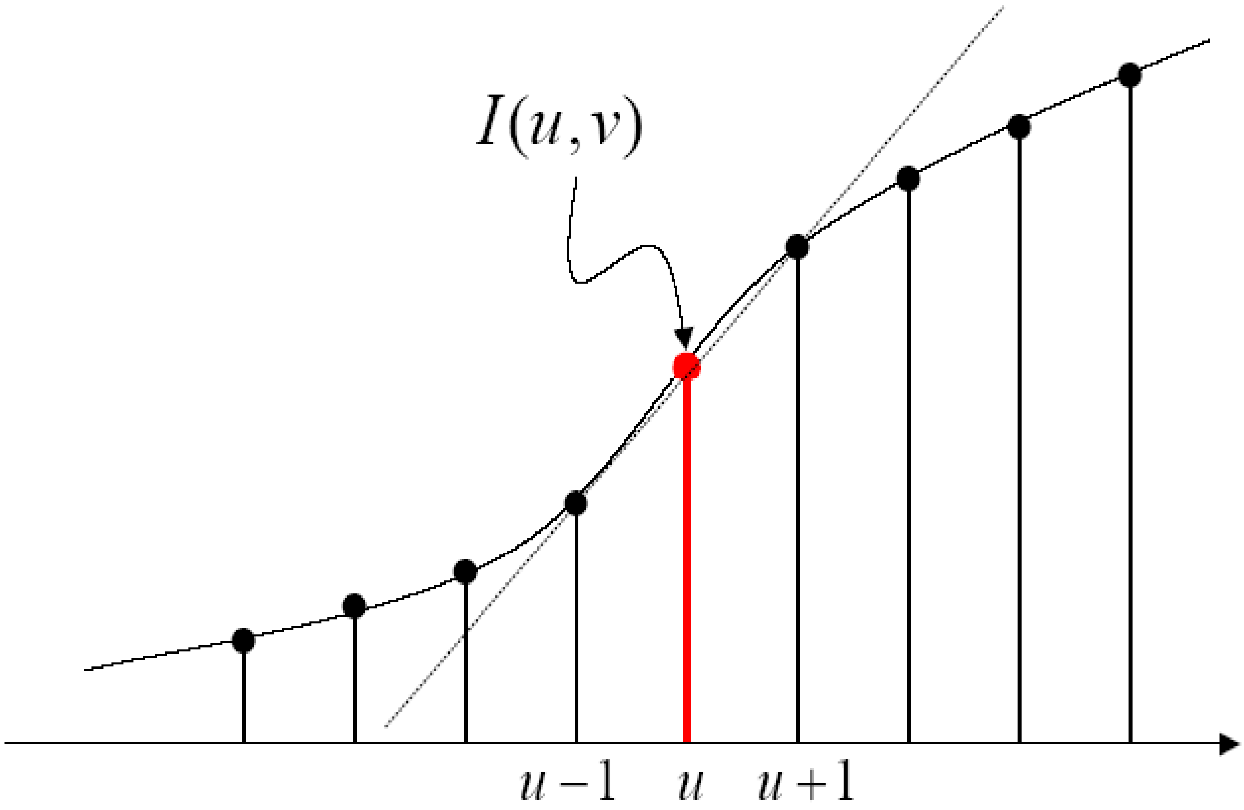
What happens at an edge?





$$\frac{\partial I(u,v)}{\partial u}$$

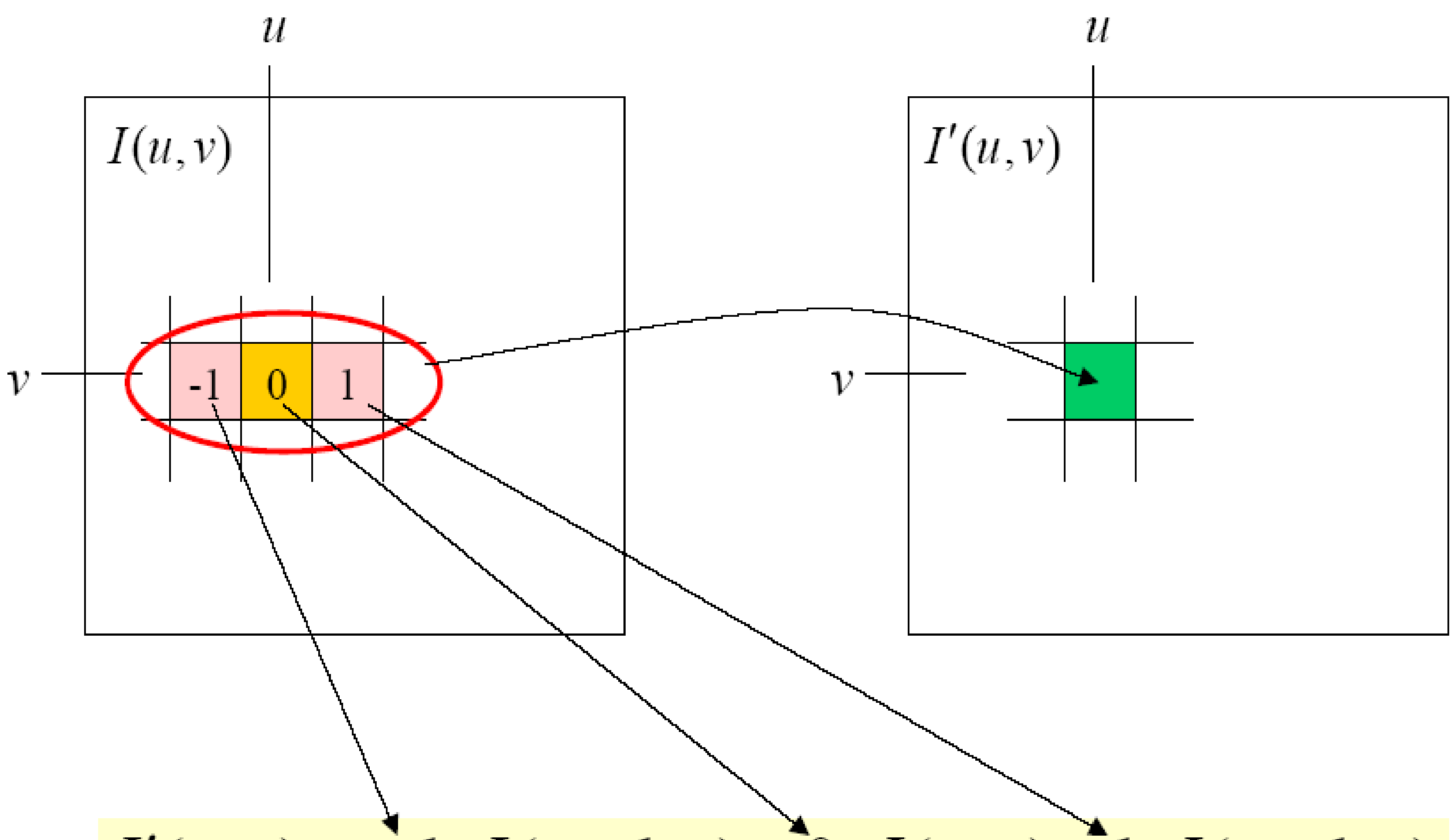
How to measure the 1st Derivative?



$$\frac{\partial I(u,v)}{\partial u} \approx I(u+1,v) - I(u-1,v)$$

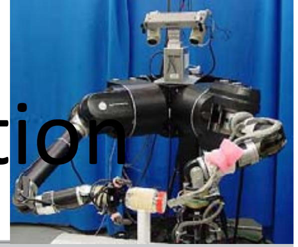
for discrete signals only approximation possible

Simple horizontal Contour Filter



$$I'(u,v) = -1 \cdot I(u-1,v) + 0 \cdot I(u,v) + 1 \cdot I(u+1,v)$$

Goal: Find Edges independent of their Orientation



- Method by Roberts (1965)

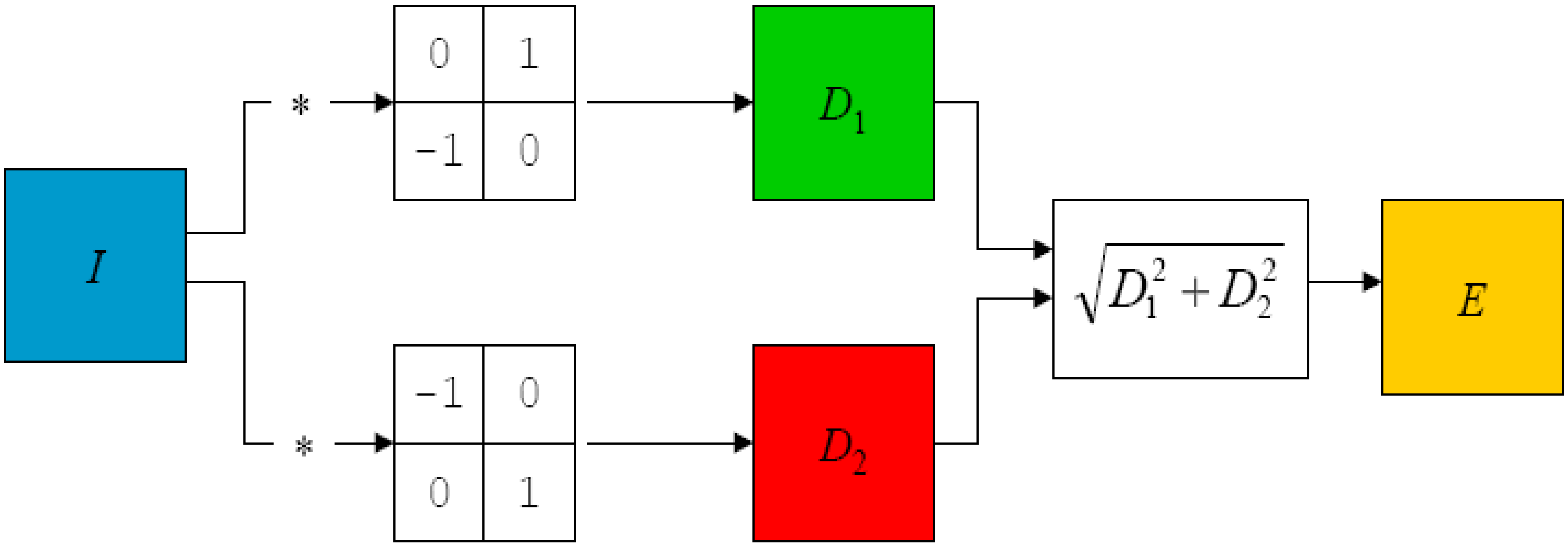
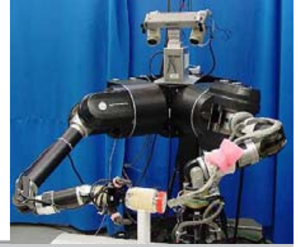
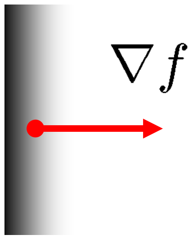


Image Gradient

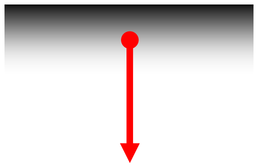


- The gradient of an image: $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$

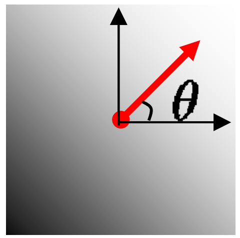
- The gradient points in the direction of most rapid increase in intensity



$\nabla f = \left[\frac{\partial f}{\partial x}, 0\right]$



$\nabla f = \left[0, \frac{\partial f}{\partial y}\right]$

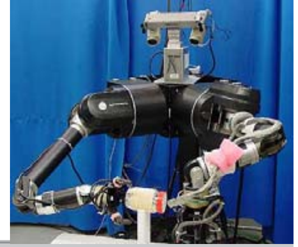


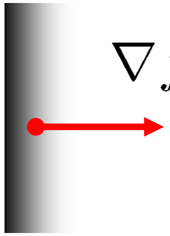
$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$

- The *edge strength* is given by the gradient magnitude:

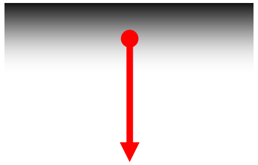
$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

Image Gradient

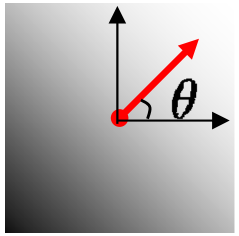




$$\nabla f = \left[\frac{\partial f}{\partial x}, 0\right]$$



$$\nabla f = \left[0, \frac{\partial f}{\partial y}\right]$$



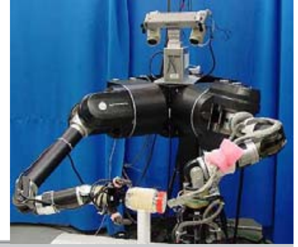
$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

- The gradient points in the direction of most rapid increase in intensity

$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$$

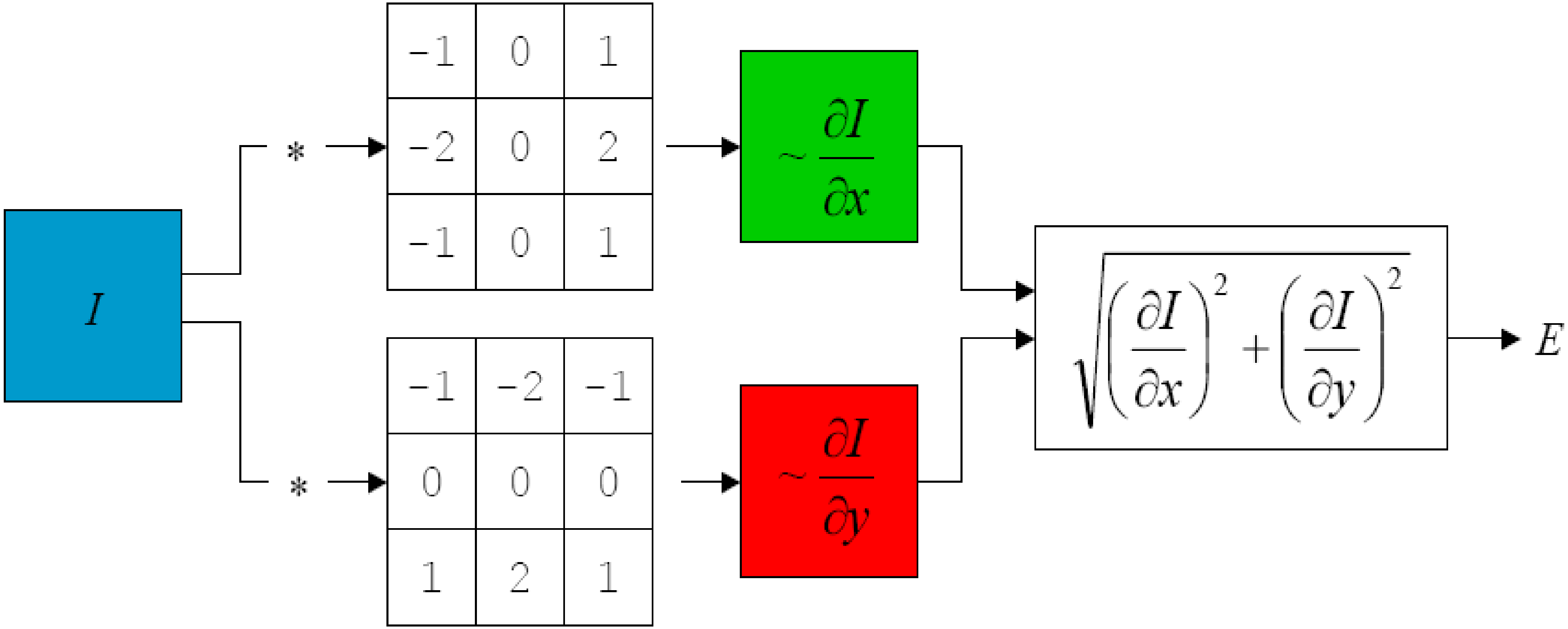
- **The gradient of a surface at a point defines the tangential plane to the surface at this point**

- how does this relate to the direction of the edge?

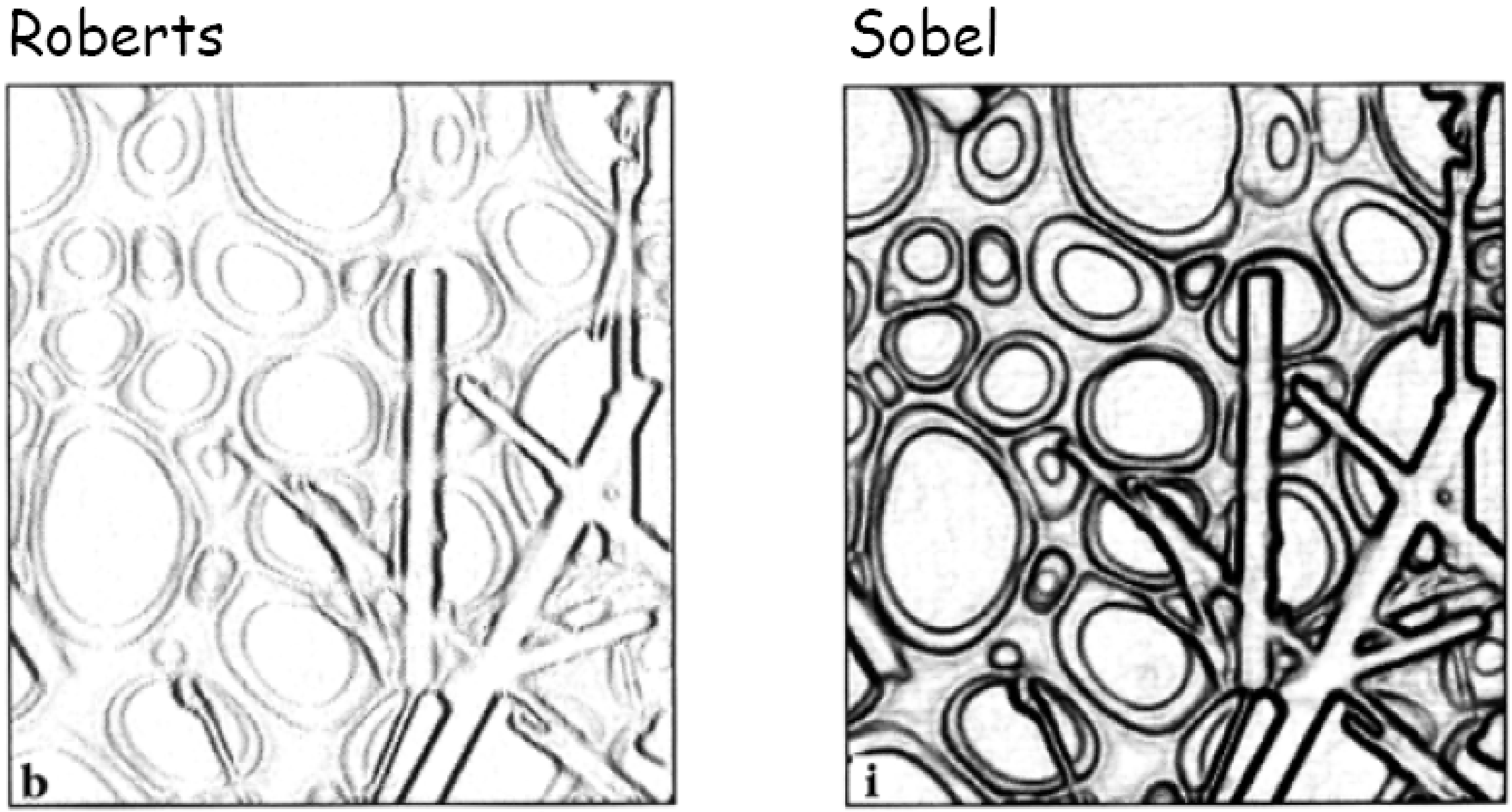


Sobel Filter

- Less noise dependent than Robert's (due to bigger filter size)



Roberts vs. Sobel Operator



Aus: John C. Russ, *The Image Processing Handbook*, CRC Press (1998)

Related Operators

δx $\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ δy $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

2x2 Roberts

δx $\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$ δy $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$

3x3 Prewitt

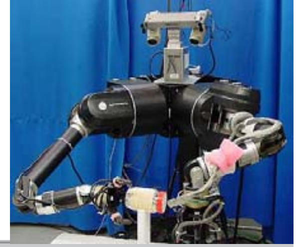
δx $\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$ δy $\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$

3x3 Sobel

δx $\begin{bmatrix} -3 & -1 & 1 & 3 \\ -3 & -1 & 1 & 3 \\ -3 & -1 & 1 & 3 \\ -3 & -1 & 1 & 3 \end{bmatrix}$ δy $\begin{bmatrix} 3 & 3 & 3 & 3 \\ 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 \\ -3 & -3 & -3 & -3 \end{bmatrix}$

4x4 Prewitt

Kirsch-Operator (1971)



- Additional filters (different orientations)
- Less orientation dependent than Sobel-Operator

D_1 (→)

−1	0	1
−2	0	2
−1	0	1

D_2 (↘)

−2	−1	0
−1	0	1
0	1	2

D_3 (↓)

−1	−2	−1
0	0	0
1	2	1

D_4 (↙)

0	−1	−2
1	0	−1
2	1	0

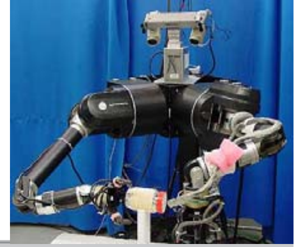
$$\max(D_1, D_2, D_3, D_4)$$

↓

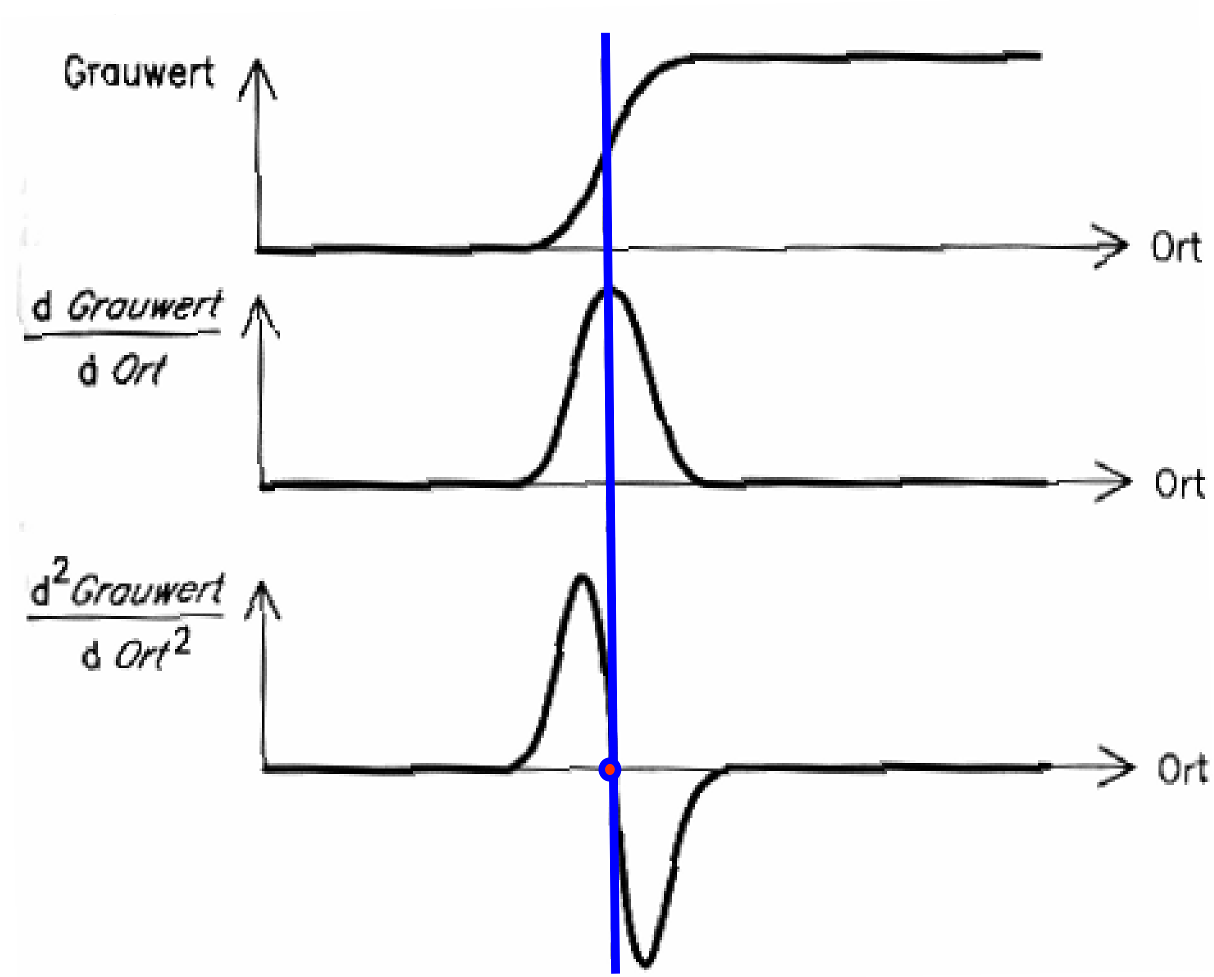
E

No square root!
„Compass“ Operator

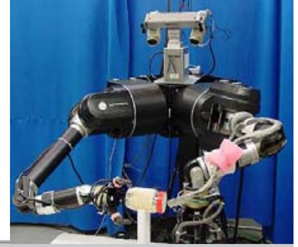
Zero Crossing



To estimate the position of the edge more precisely the zero crossing of the 2nd derivative is determined



Laplace Filter

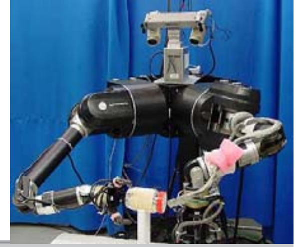


- Mathematical approximation of second derivative. Drawback: **not directional anymore!**

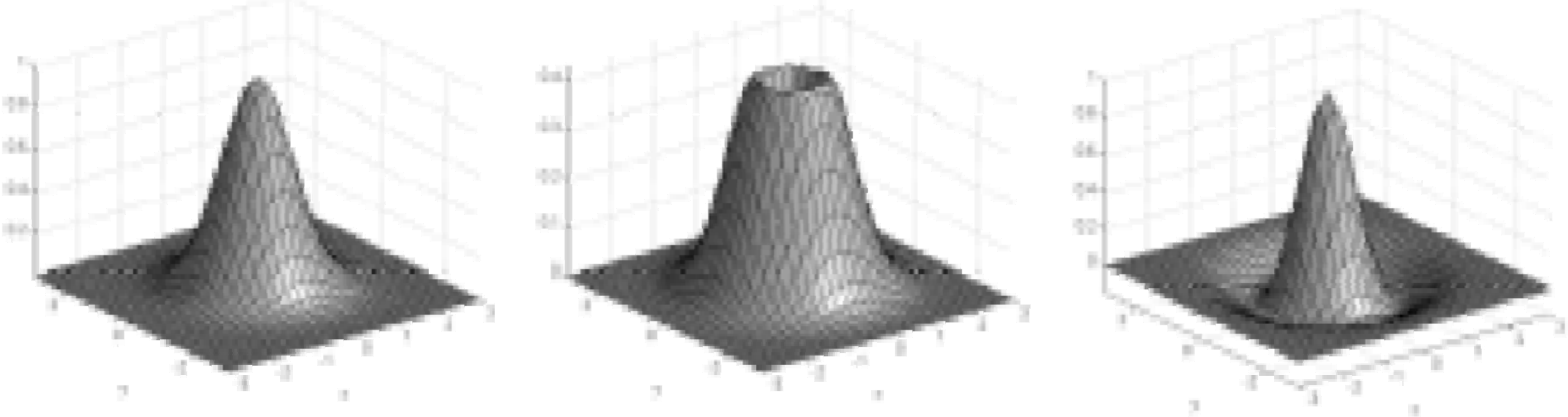
$$L = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \text{or} \quad L = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

- Because of the high pass characteristics (2nd order filter = 2nd derivative) the Laplace filter is **very sensitive to noise**.
- Therefore it is rarely applied alone. Usually it is combined with a Gaussian Filter that reduces noise before the Laplace filter can be applied.

Laplace of Gaussian (LoG)

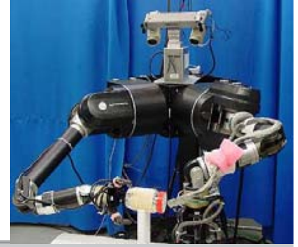


- Combination of **Gaussian Filter** and **Laplace Filter**
- Combination corresponds to second derivative of the 2D Gaussian function Laplacian of Gaussian filter (LoG):
  - Because of the shape of its kernel elements the LoG filter is usually called "**Mexican Hat**" filter
  - LoG Filter can be used for Edge Detection
  - LoG Filter **does not depend** on a particular direction.



Gauss' Bell Function | First derivative | Minus second derivative „Mexican Hat"

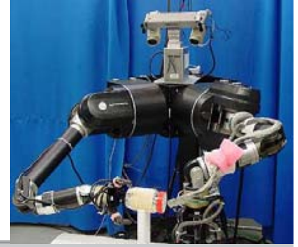
Canny Edge Detector

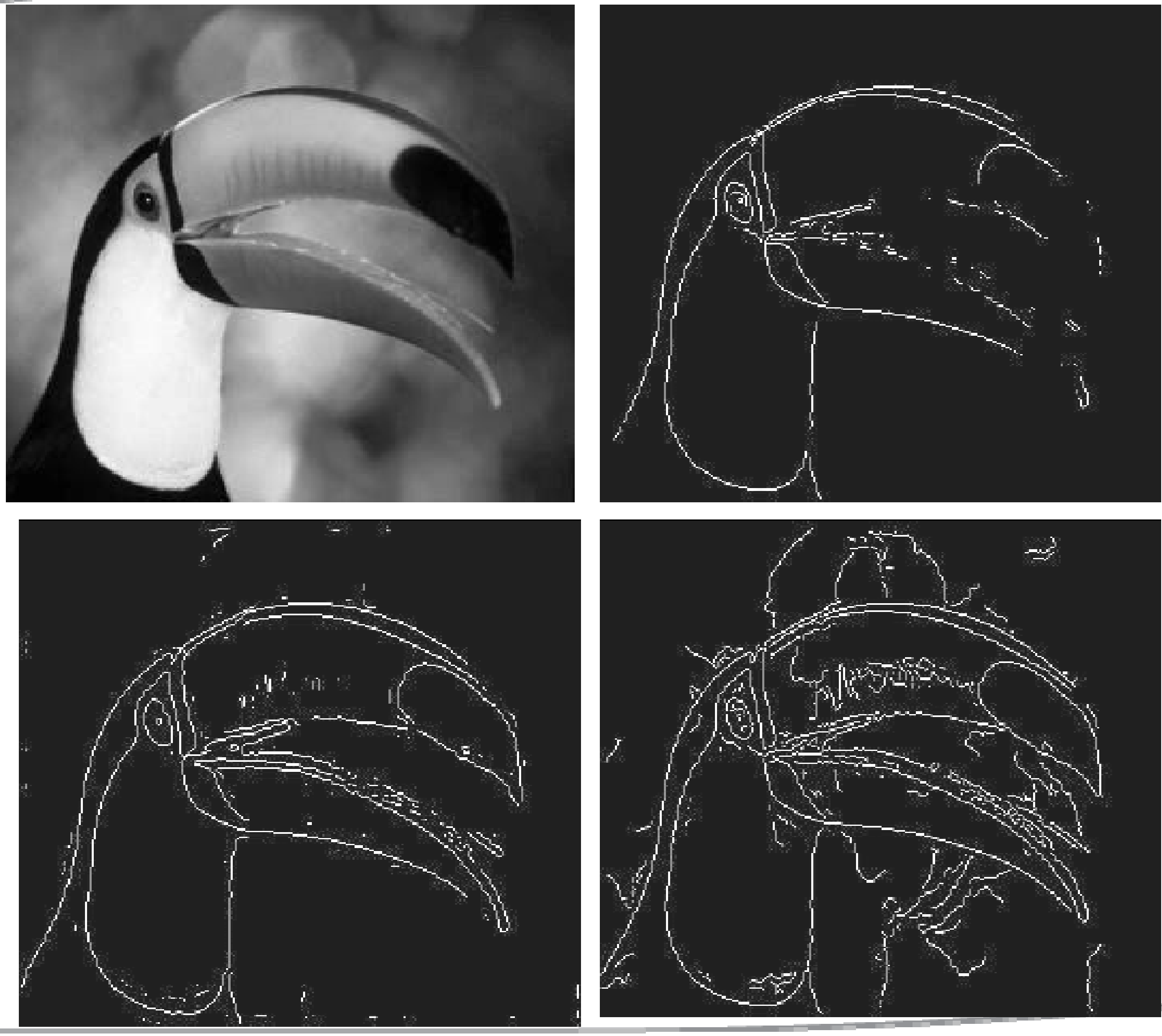


MATLAB: `edge(image,'canny')`

1. Filter image with derivative of Gaussian
2. Find magnitude and orientation of gradient
3. Non-maximum suppression
4. Linking and thresholding (hysteresis):
   - Define two thresholds: low and high
   - Use the high threshold to start edge curves and the low threshold to continue them

Example





Sobel

LOG

Canny

From Edgels to Edges