Department of Computer Science
University of Bristol

Image Processing and Computer Vision

www.ole.bris.ac.uk/bbcswebdav/courses/COMS30121_2018/content
www.ole.bris.ac.uk/bbcswebdav/courses/COMSM0020_2018/content


(First Ever) Photo by J N Niépce

Lecture 02

Image Acquisition & Representation

Sion Hannuna | Tilo Burghardt | Andrew Calway

18 Slides

Agenda: Basics of Image Acquisition and Representation

Images as Sensory Data

- How are images acquired?
- Which processes influence digital image formation?

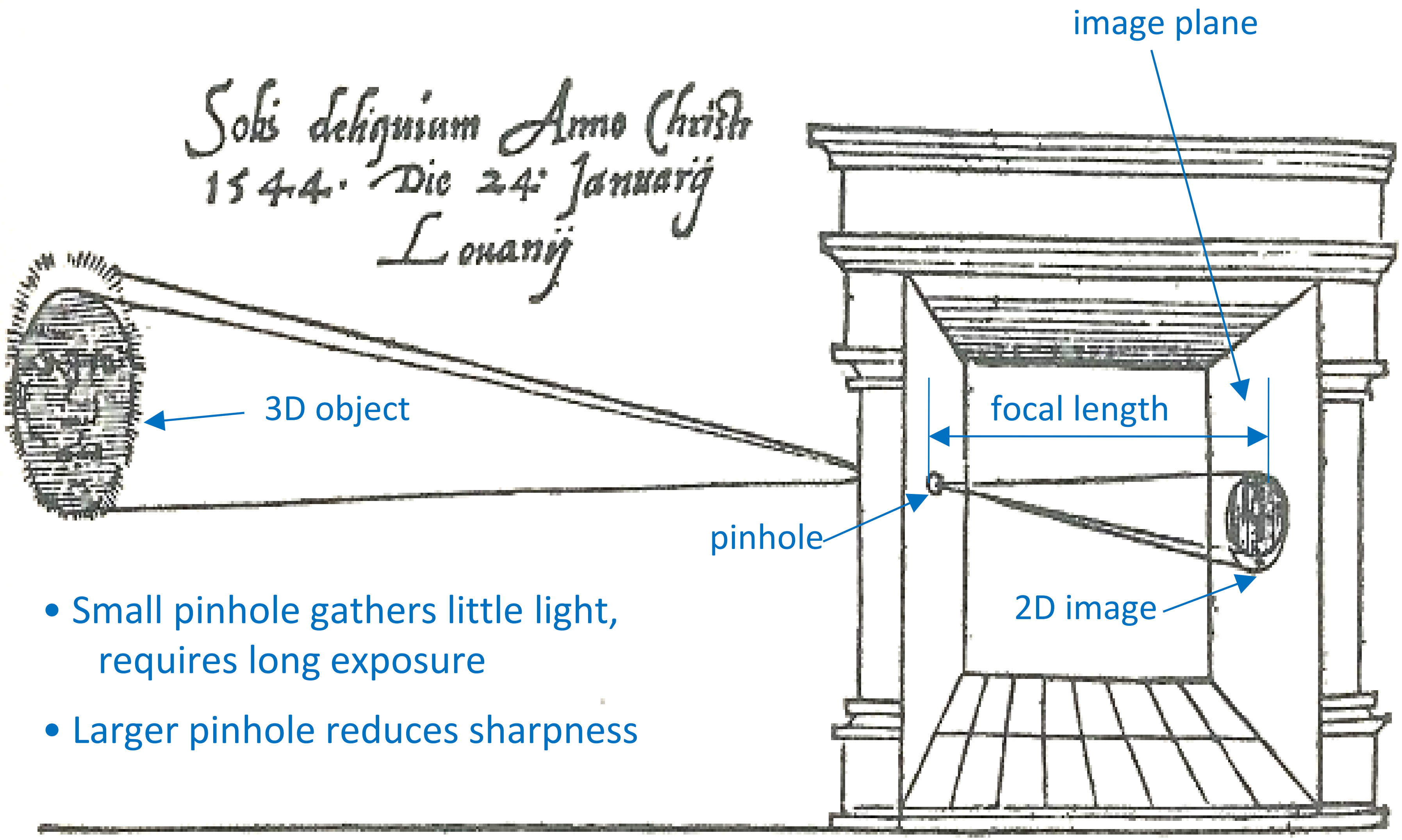
Images as Structured Data

- How can digital images be represented?
- How are digital images altered by acquisition and representation?

Manual Perspective: 3D → 2D Projection

Part of the ‚Perspective Machine', by Albrecht Dürer (1525)

The Camera Obscura (Pinhole Camera)

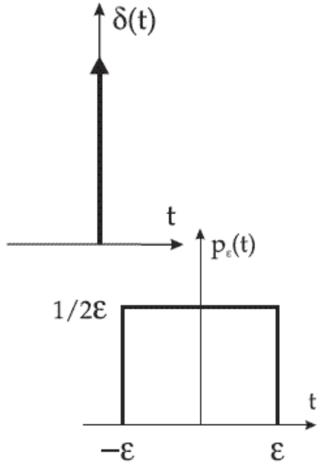


- Small pinhole gathers little light, requires long exposure
- Larger pinhole reduces sharpness

First published picture of camera obscura in Gemma Frisius' 1545 book *De Radio Astronomica et Geometrica*

Images as Accumulations of Scaled Dirac Delta-Functions

Definition $$\int_{-\infty}^{\infty} \delta(t)dt = 1$$

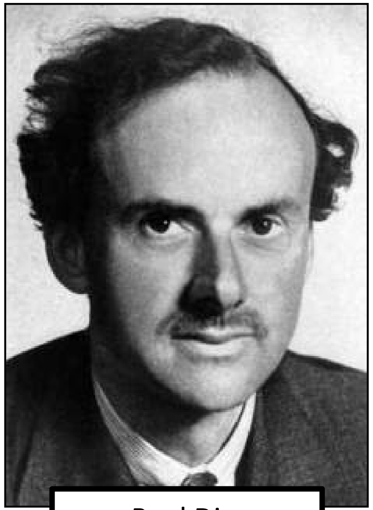


More intuitively $$\delta(t) = \lim_{\varepsilon \to 0} [p_\varepsilon(t)]$$

Shifting property

$$\int_{-\infty}^{\infty} f(t)\delta(t)dt = f(0)$$

$$\int_{-\infty}^{\infty} f(t)\delta(t-t_0)dt = f(t_0)$$

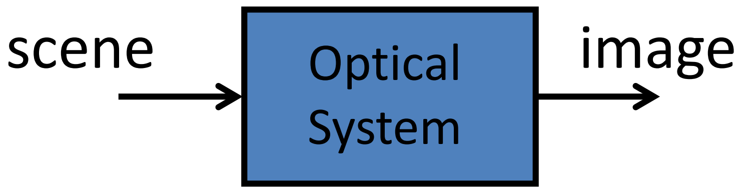


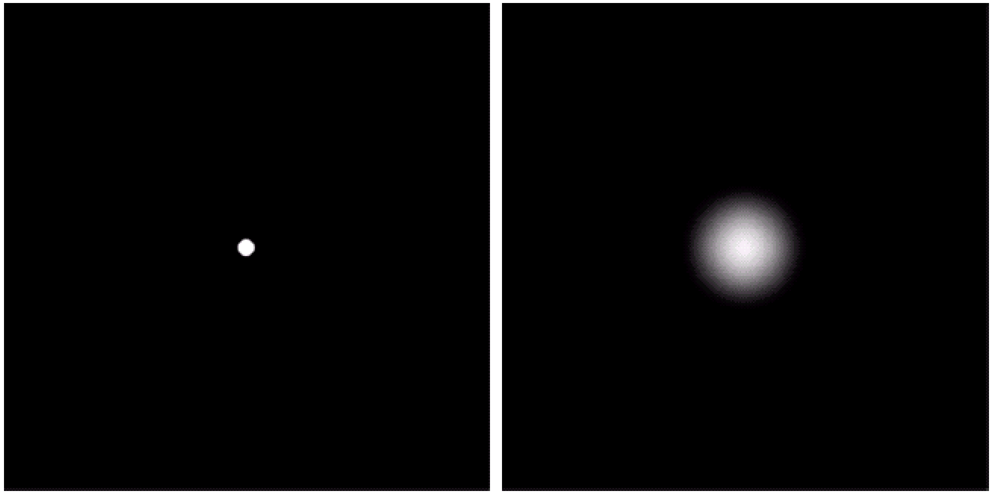
Paul Dirac

The shifting property can be used to express a 2D ‚image function' as a linear combination of scaled 2D Dirac pulses located at points (a,b) that cover the whole image plane:

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(a,b)\, \delta(a-x, b-y)\, da\, db = f(x,y)$$

The Point Spread Function





- Ideally, the optical system should be mapping point information to points again.

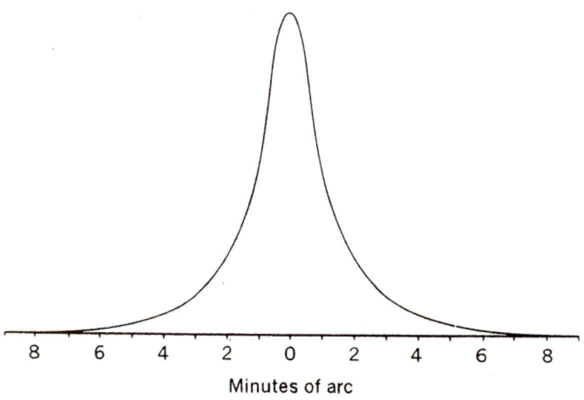
- However, optical systems are never ideal.

$$\delta(x) \longrightarrow \boxed{\text{Optical System}} \longrightarrow PSF(x)$$

point source — point spread function

- Superposition Principle:
  An image is the sum of the PSF of all its points.

  - Point spread function of Human Eyes

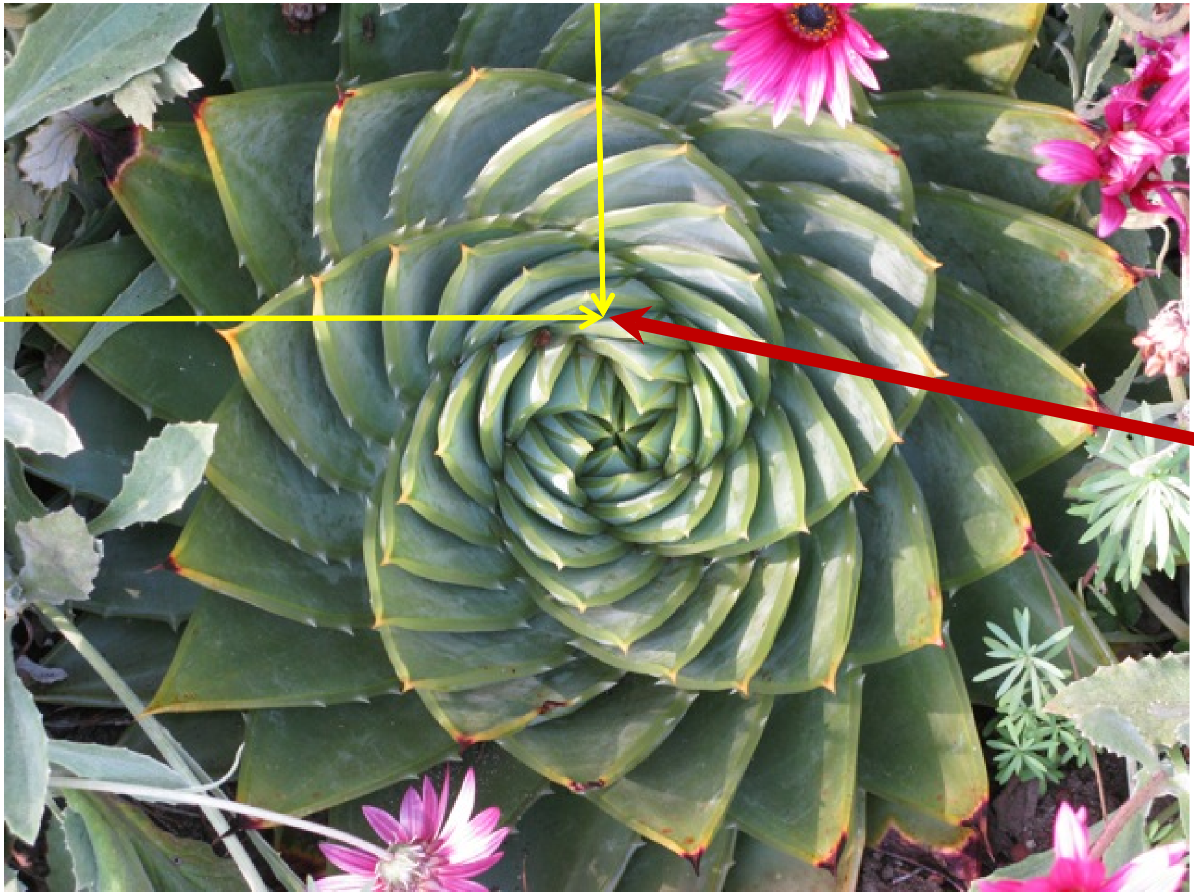


How to Model a Continuous Image Function f ?

localisation

x_1

x_2



(c_R, c_G, c_B)

colouration

Space of Continuous Image Functions f

- Continuous image function $\boldsymbol{f}$ that maps from coordinates $\boldsymbol{x_1, x_2, \ldots}$ to (colour) values $\boldsymbol{c_1, c_2, \ldots}$

$$f(x_1, x_2, \ldots, x_m) = (c_1, c_2, \ldots, c_n)$$

$$f : \Re^m \rightarrow \Re^n$$

m … image dimensionality $\quad$ n … channels

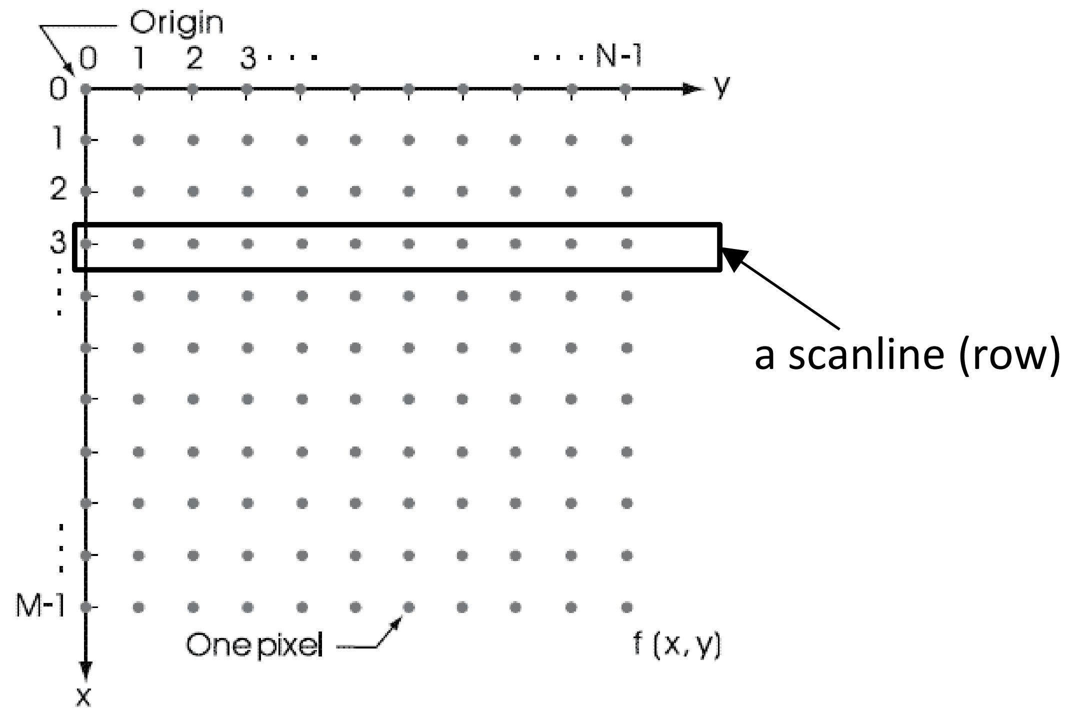
- Specific Example: image sequences, e.g. video

$$f(x, y, t) = 128$$

(x, y) are image coordinates and t is the frame number.

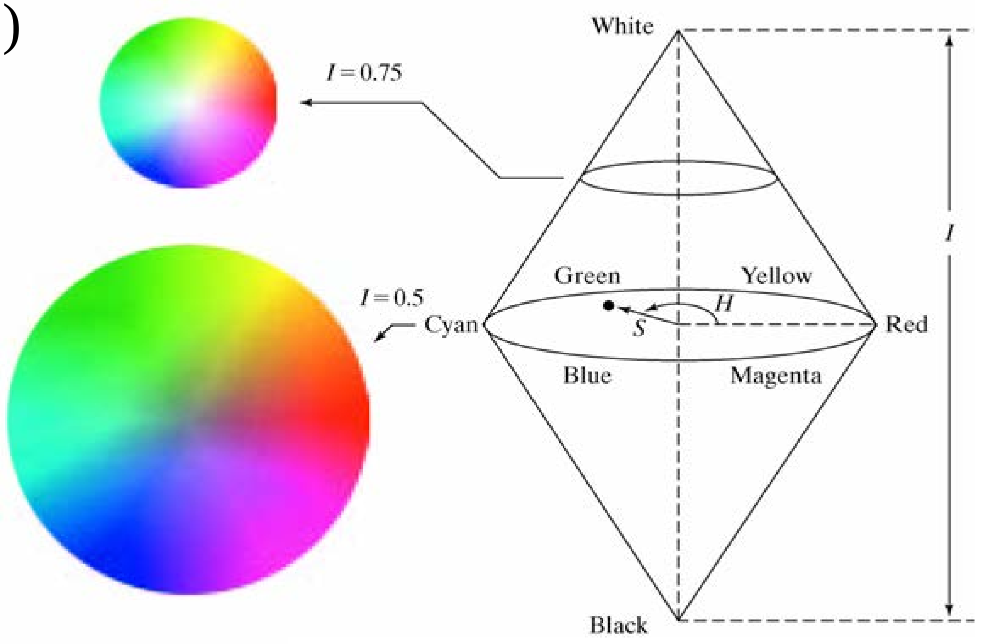
How to Represent a (Spatially Discrete) Image?

$$f[x,y] = \begin{bmatrix} f[0,0] & f[0,1] & \cdots & f[0,N-1] \\ f[1,0] & f[1,1] & \cdots & f[1,N-1] \\ \vdots & \vdots & \ddots & \vdots \\ f[M-1,0] & f[M-1,1] & \cdots & f[M-1,N-1] \end{bmatrix}$$

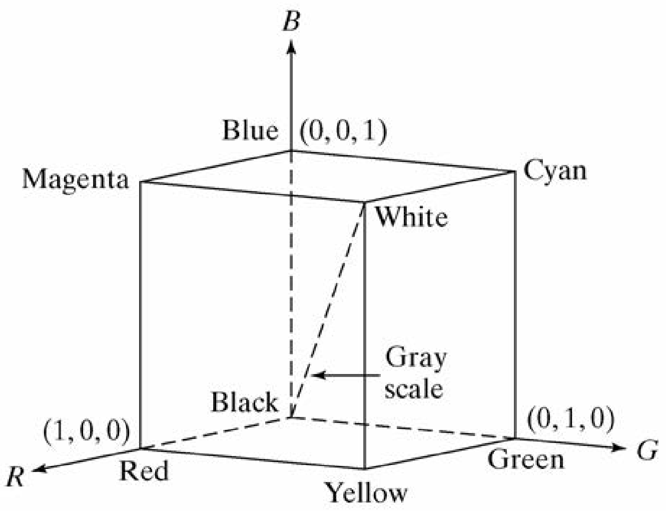


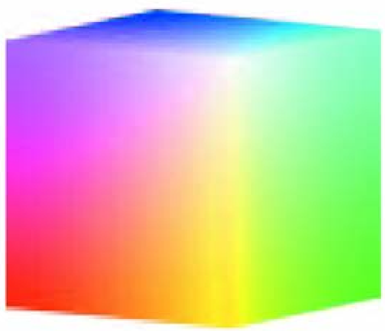
Representation of Colour Spaces

$$f(x,y) = (H,S,I)$$



$$f(x,y) = (R,G,B)$$





Other examples are:

$$f(x,y) = (L,a,b)$$
$$f(x,y) = (Y,U,V)$$
$$\ldots$$

Example: OpenCV(C++) Image Creation

draw.cpp

```
#include [...]
using namespace cv;

int main() {
  //create a red 256x256, 8bit, 3channel BGR image in a matrix container
  Mat image(256, 256, CV_8UC3, Scalar(0, 0, 255));
```

width & height

8 bits per channel

unsigned

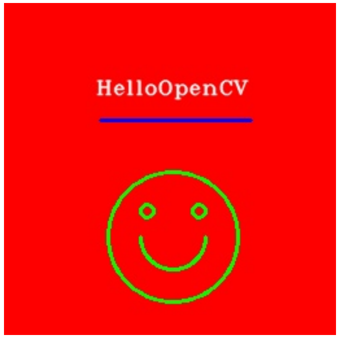
3 channels

blue

green

red

```
  //put white text HelloOpenCV
  putText(image, "HelloOpenCV", Point(70, 70),
    FONT_HERSHEY_COMPLEX_SMALL, 0.8, cvScalar(255, 255, 255), 1, CV_AA);
  //draw blue line under text
  line(image, Point(74, 90), Point(190, 90), cvScalar(255, 0, 0),2);
  //draw a green smile
  ellipse(image, Point(130, 180), Size(25,25), 180, 180, 360,
    cvScalar(0, 255, 0), 2);
  circle(image, Point(130, 180), 50, cvScalar(0, 255, 0), 2);
  circle(image, Point(110, 160), 5, cvScalar(0, 255, 0), 2);
  circle(image, Point(150, 160), 5, cvScalar(0, 255, 0), 2);
  //save image to file
  imwrite("myimage.jpg", image);
  //free memory occupied by image
  image.release();
  return 0;
}
```

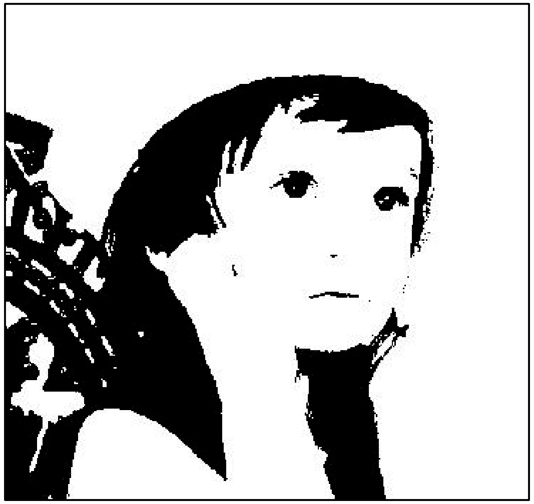


Quantization of the Image Function

Representing a continuously varying single channel image function ***f(x)*** with a discrete one using quantization levels:

16 levels

6 levels



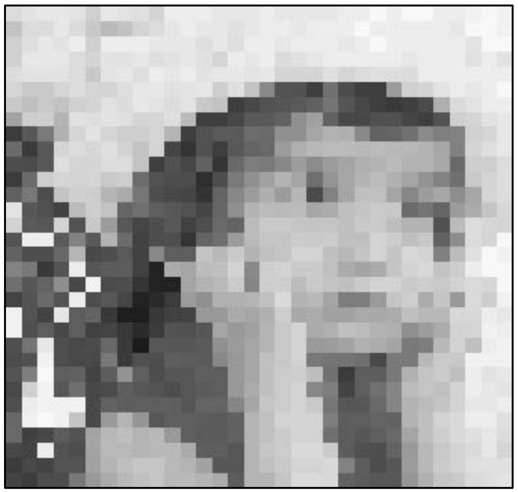
2 levels

Spatial Sampling in Practice

The effect of very sparse sampling … is often ALIASING

256 x256

64x64



32x32

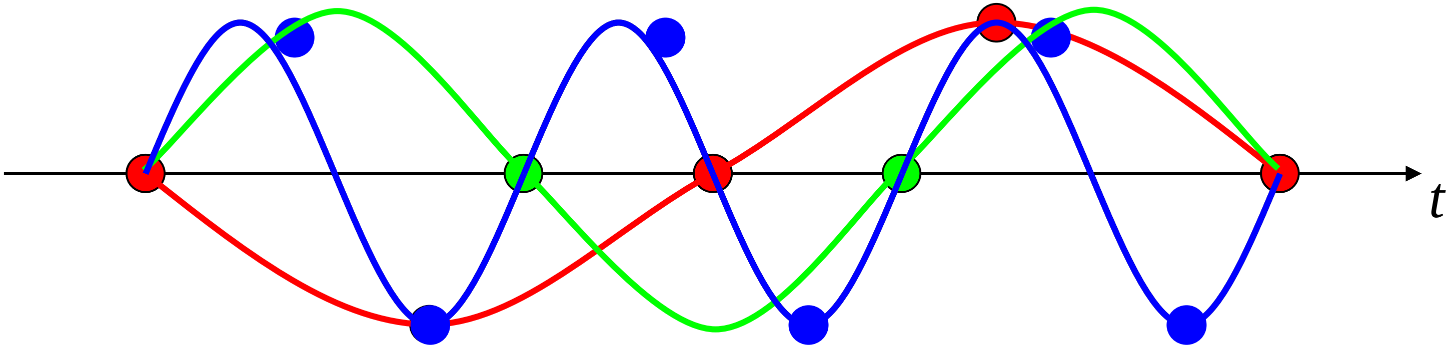
Anti-aliasing can be achieved by removing all spatial frequencies above a critical limit (so-called Shannon-Nyquist Limit).

Shannon's Sampling Theorem

"An analogue signal containing components up to some maximum frequency ***u*** may be completely reconstructed by regularly spread samples, provided the sampling rate is above 2***u*** samples per second."

Claude Shannon

Also referred to as the Shannon-Nyquist criterion: Sampling **must** be performed **above twice** the highest (spatial) frequency of the signal to be lossless.

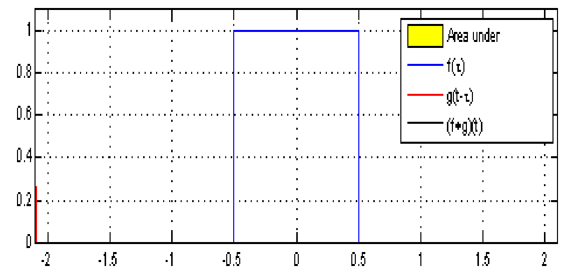


Convolution

- ...quantifies the structural similarity of a kernel image ***h(x)*** as it is shifted over a target image ***f(x)***:

$$f * h = \int_{-\infty}^{+\infty} f(x-t)h(x)\,\partial t$$

Convolution symbol



Animation Source: Brian Amberg

- ... determines the effect of a system, i.e. the kernel ***h(x)***, on an input signal, i.e. ***f(x)***
- the result image is known as the \`response' of ***f*** to the kernel ***h***

2D Discrete Convolution

- The discrete version of 2D convolution is defined as:

$$g(x,y) = \sum_{m}\sum_{n} f(x-m, y-n)h(m,n)$$

$$\underbrace{\begin{array}{c|c|c|c|c|c}
 & & y-1 & y & y+1 & \\
 & & & & & \\
x-1 & & 43 & 12 & 61 & \\
x & & 44 & \boxed{45} & 60 & \\
x+1 & & 43 & 50 & 61 & \\
 & & & & &
\end{array}}_{f} \; * \; \underbrace{\begin{array}{c|c|c|c}
 & -1 & 0 & 1 \\
-1 & -1 & 0 & 1 \\
0 & -2 & 0 & 2 \\
1 & -1 & 0 & 1
\end{array}}_{h} = -68 =$$

$$\begin{aligned}
& f(x+1, y+1)h(-1,-1) \\
+ & f(x+1, y)h(-1,0) \\
+ & f(x+1, y-1)h(-1,1) \\
+ & f(x, y+1)h(0,-1) \\
+ & f(x, y)h(0,0) \\
+ & f(x, y-1)h(0,1) \\
+ & f(x-1, y+1)h(1,-1) \\
+ & f(x-1, y)h(1,0) \\
+ & f(x-1, y-1)h(1,1)
\end{aligned}$$

2D Discrete Correlation

- The discrete version of 2D correlation is defined as:

$$g(x,y) = \sum_{m}\sum_{n} f(x+m, y+n)h(m,n)$$

	-1	0	1
-1	**-1**	**0**	**1**
0	**-2**	**0**	**2**
1	**-1**	**0**	**1**

		$y-1$	y	$y+1$	
$x-1$		**43**	**12**	**61**	
x		**44**	**45**	**60**	
$x+1$		**43**	**50**	**61**	

68

Correlation ≡ Convolution when kernel is symmetric under 180° rotation, e.g.

1	**2**	**1**

Spatial Low/High Pass Filtering