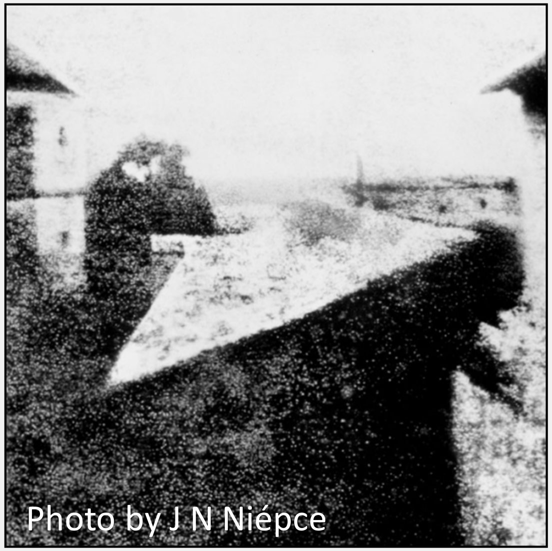
Department of Computer Science
University of Bristol

COMS30121/COMSM0020 - Image Processing and Computer Vision

www.ole.bris.ac.uk/bbcswebdav/courses/COMS30121_2017/content



Lecture 02

Image Acquisition & Representation

Andrew Calway | andrew@cs.bris.ac.uk
Tilo Burghardt | tilo@cs.bris.ac.uk
Majid Mirmehdi | majid@cs.bris.ac.uk

42 Slides

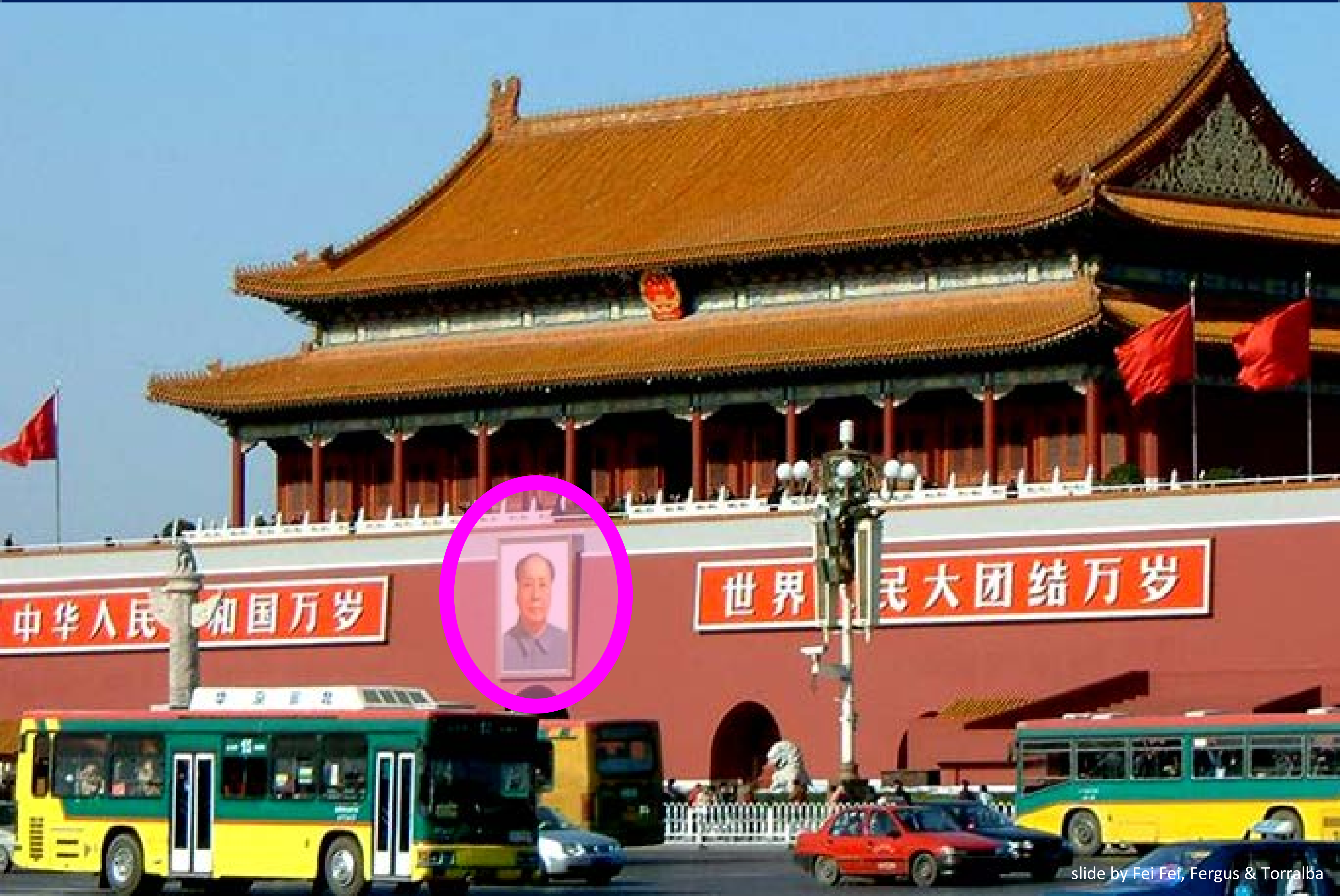
So what does the human visual system see?

slide by Fei Fei, Fergus & Torralba

Detection: are there cars?

slide by Fei Fei, Fergus & Torralba

Identification: is that a picture of Mao?

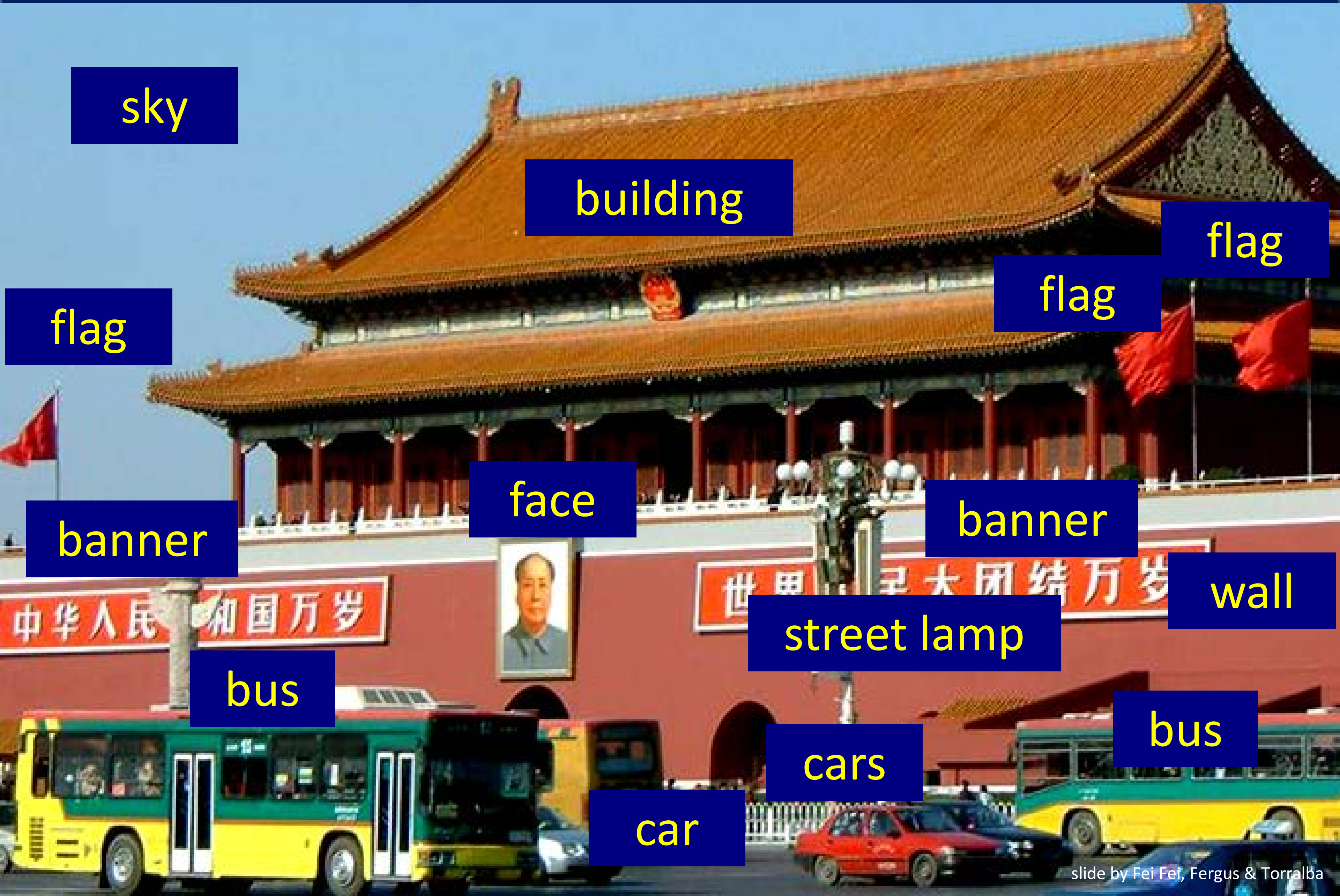


slide by Fei Fei, Fergus & Torralba

Verification: is that a bus?

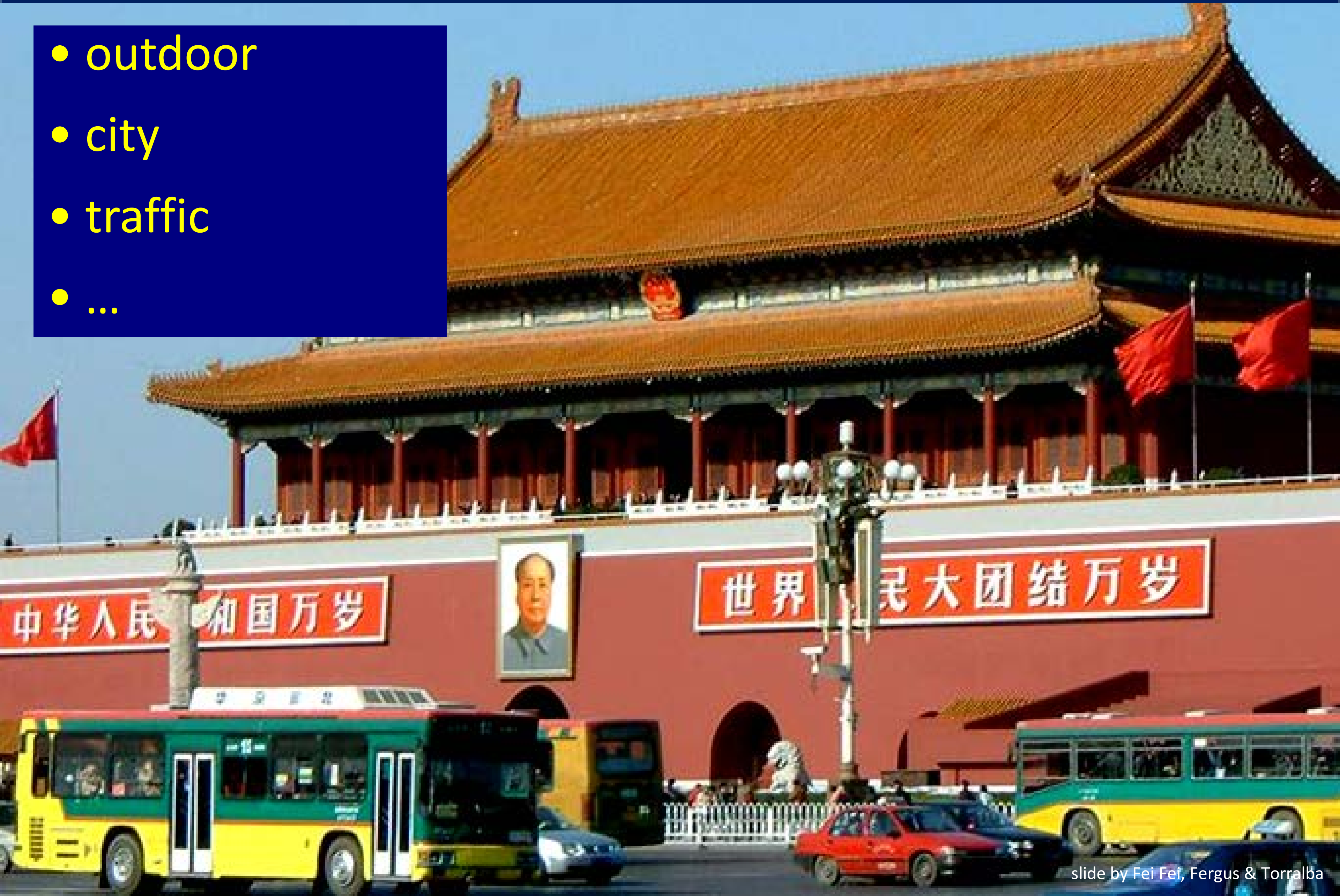
slide by Fei Fei, Fergus & Torralba

Object categorization



Scene and context categorization

- outdoor
- city
- traffic
- …



slide by Fei Fei, Fergus & Torralba

Rough 3D layout, depth ordering

slide by Fei Fei, Fergus & Torralba

Challenges

So what are the challenges faced by a *computer vision system*?

Challenge 1: view-point variation

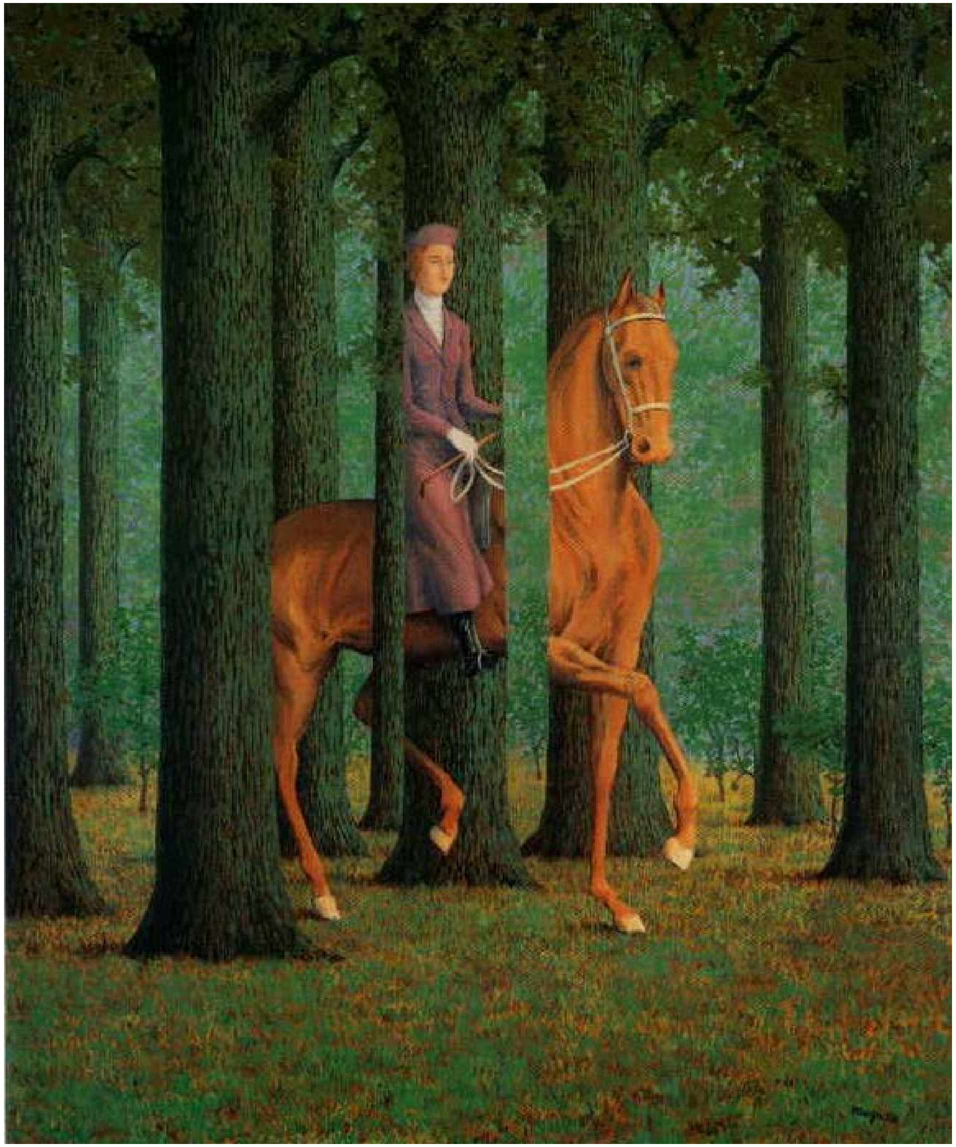
Michelangelo 1475-1564

original Challenge slides by Efros, Ullman, and others

Challenge 2: illumination

original Challenge slides by Efros, Ullman, and others

Challenge 3: occlusion



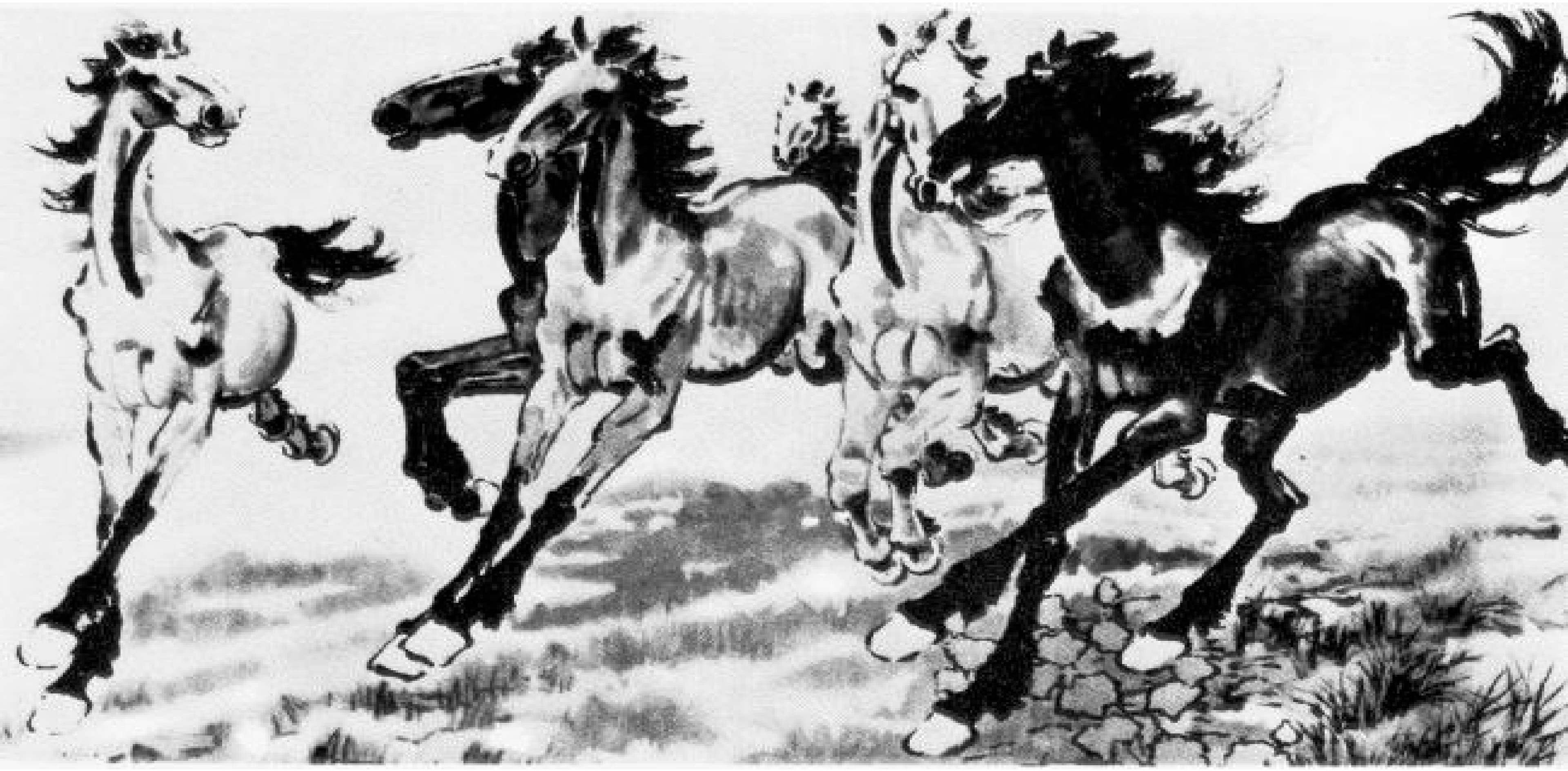
Magritte, 1957

original Challenge slides by Efros, Ullman, and others

Challenge 4: scale

original Challenge slides by Efros, Ullman, and others

Challenge 5: deformation



Xu, Beihong 1943

Challenge 6: background clutter

Klimt, 1913

original Challenge slides by Efros, Ullman, and others

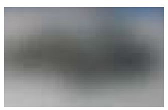
Challenge 7: object intra-class variation






original Challenge slides by Efros, Ullman, and others

Challenge 8: local ambiguity



original Challenge slides by Efros, Ullman, and others

Challenge 9: the world behind the image

The Basics of Image Acquisition and Representation

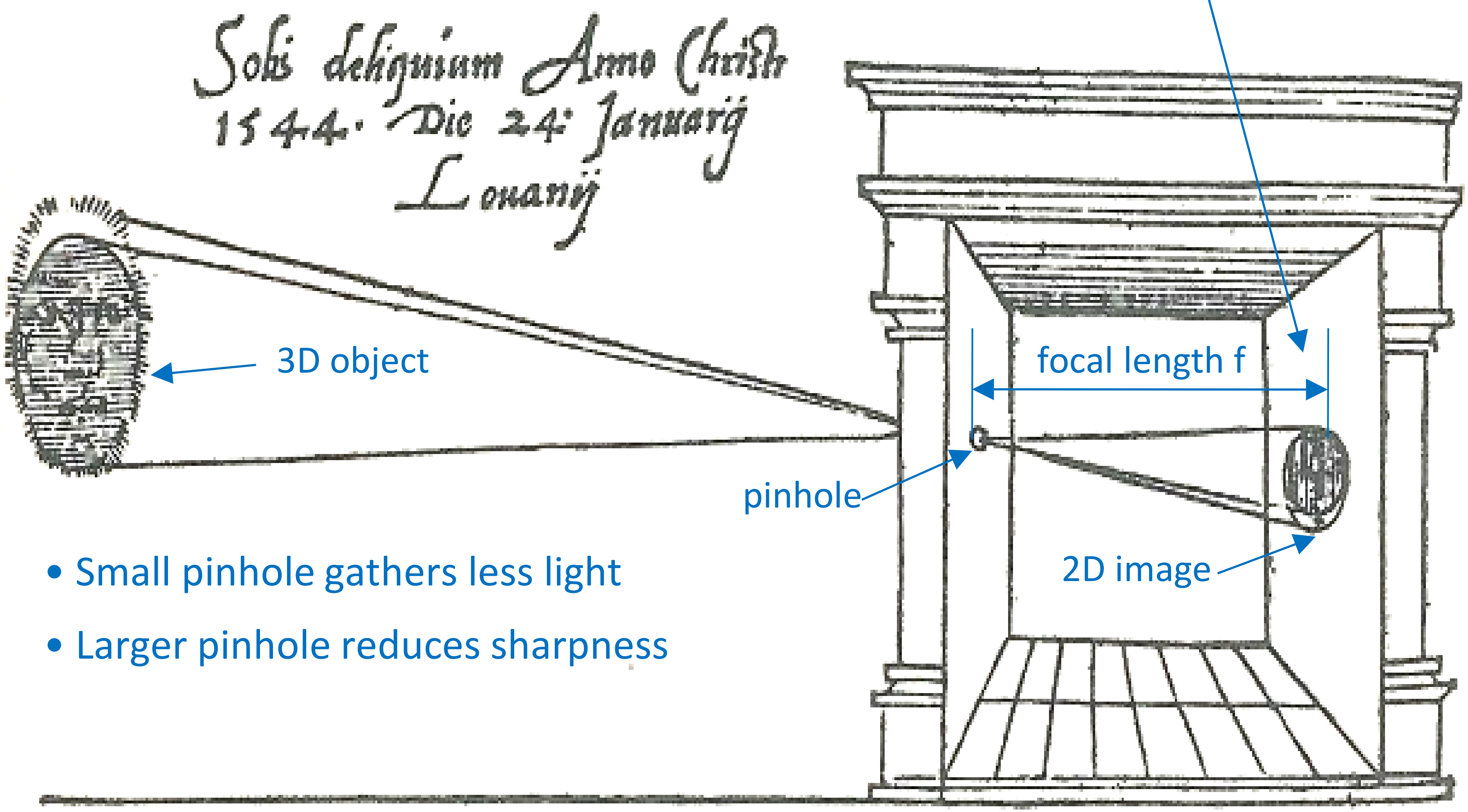
Images as Sensory Data

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

Images as Structured Data

- How can digital images be represented?

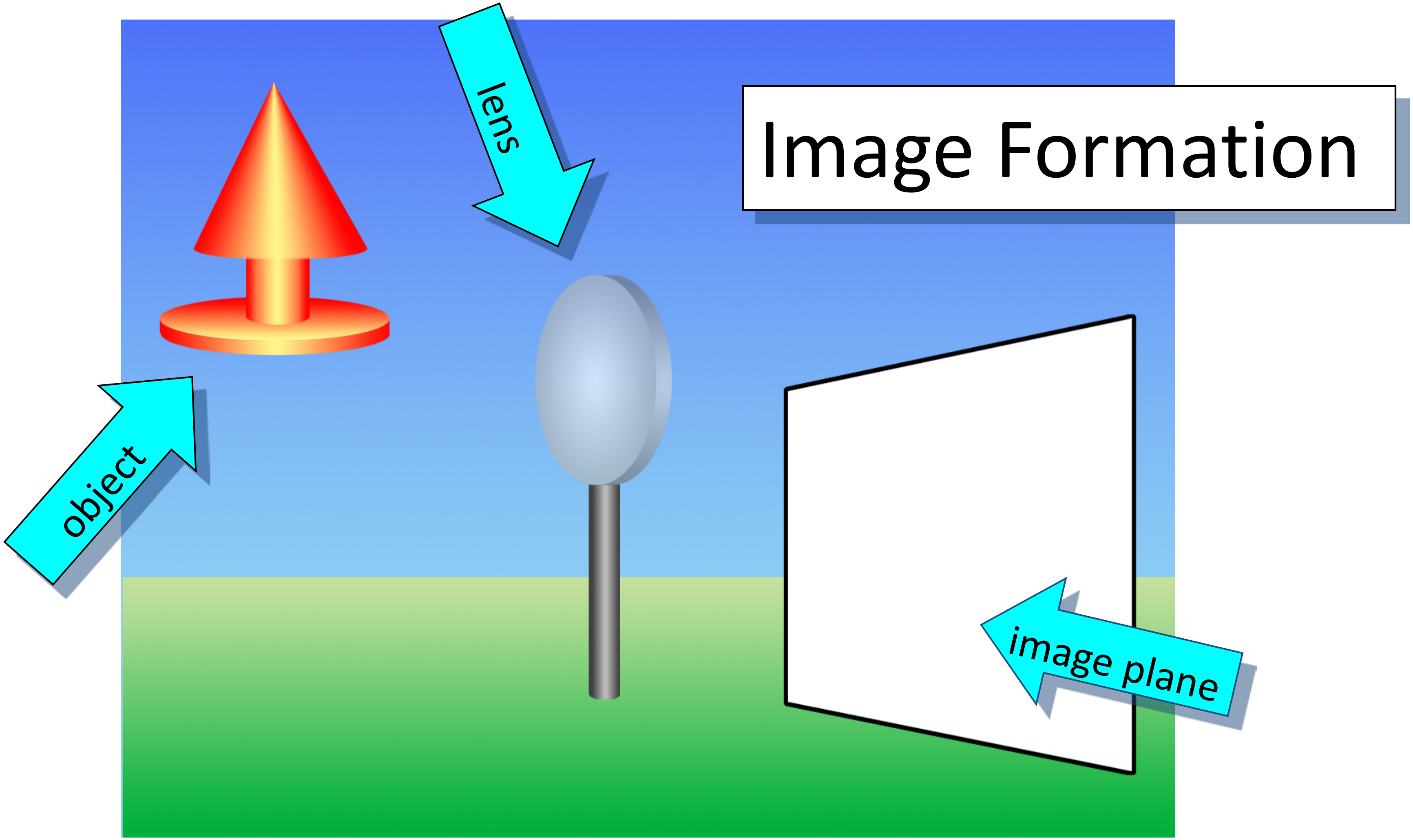
The Camera Obscura (Pinhole Camera)



- Small pinhole gathers less light
- Larger pinhole reduces sharpness

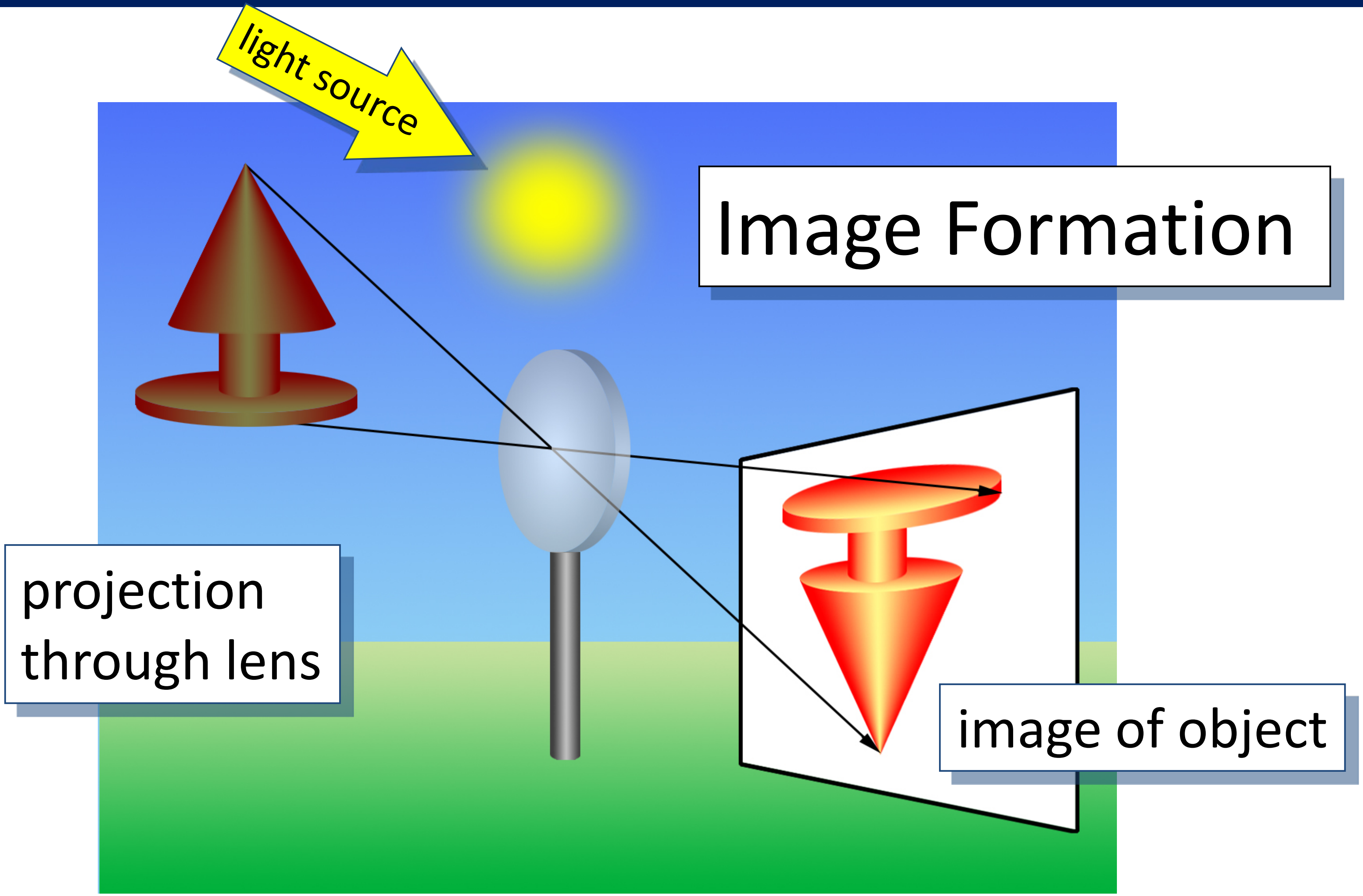
First published picture of camera obscura in Gemma Frisius' 1545 book *De Radio Astronomica et Geometrica*

Digital Image Acquisition



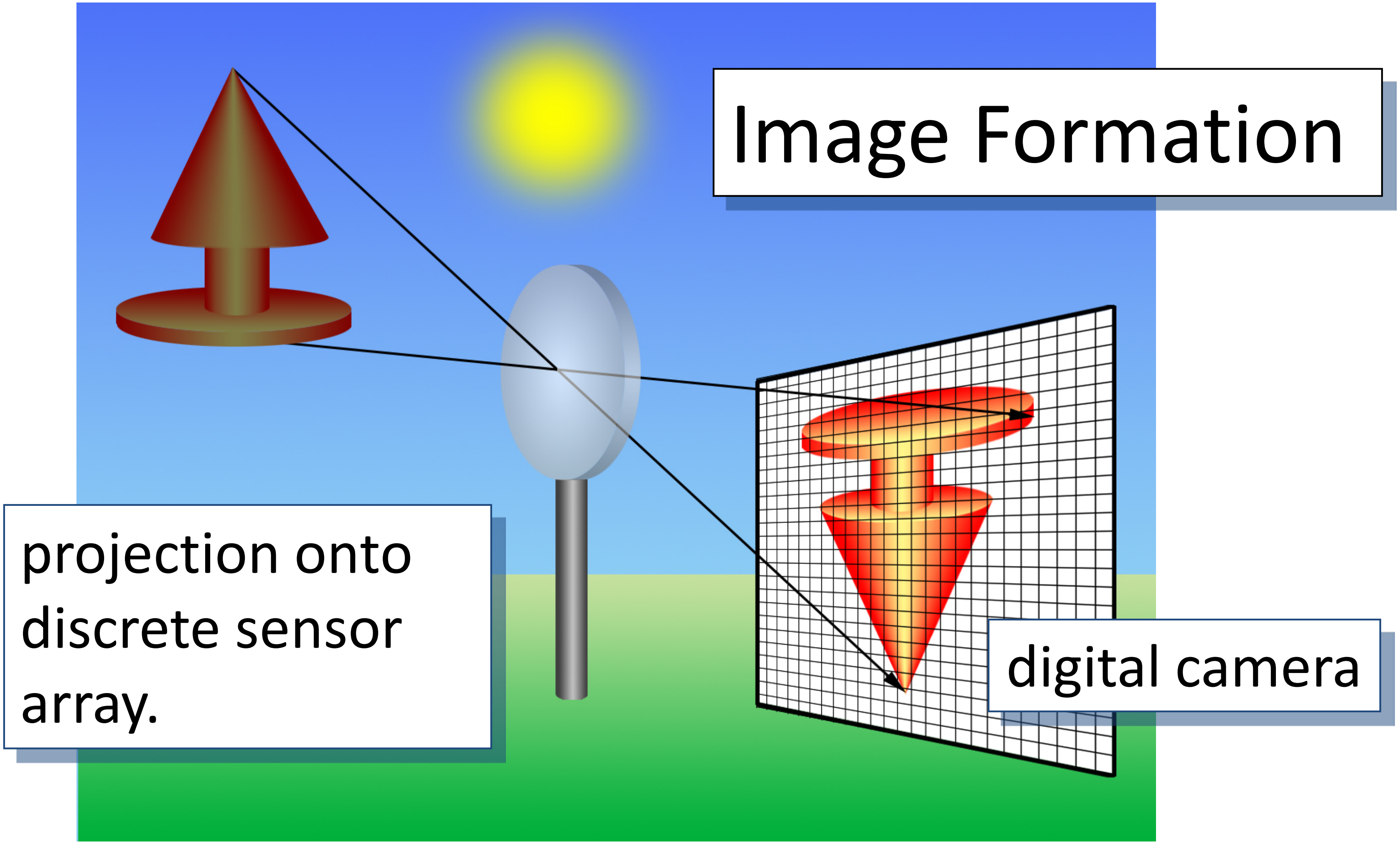
Slide by R A Peters

Digital Image Acquisition



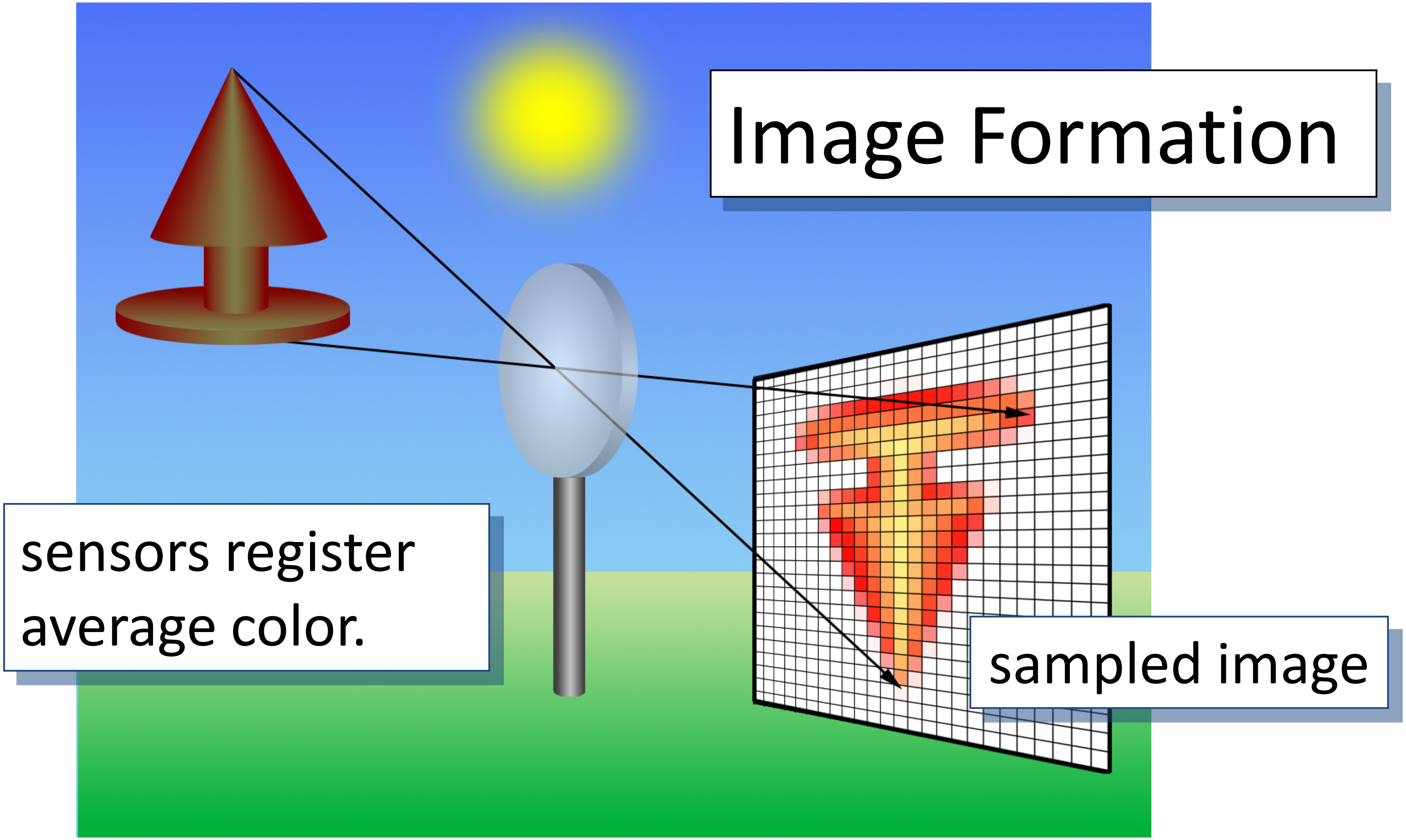
Slide by R A Peters

Digital Image Acquisition



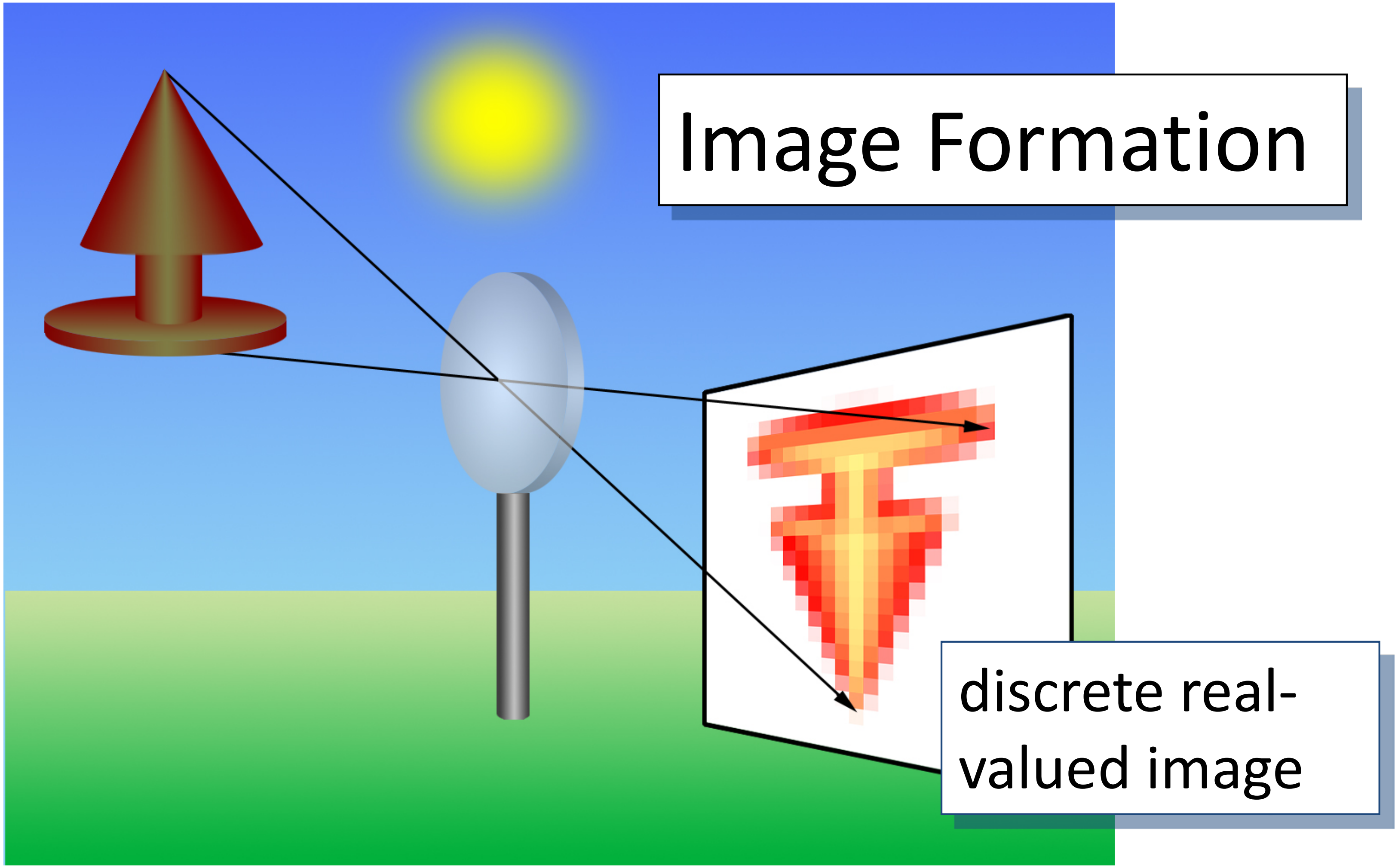
Slide by R A Peters

Digital Image Acquisition



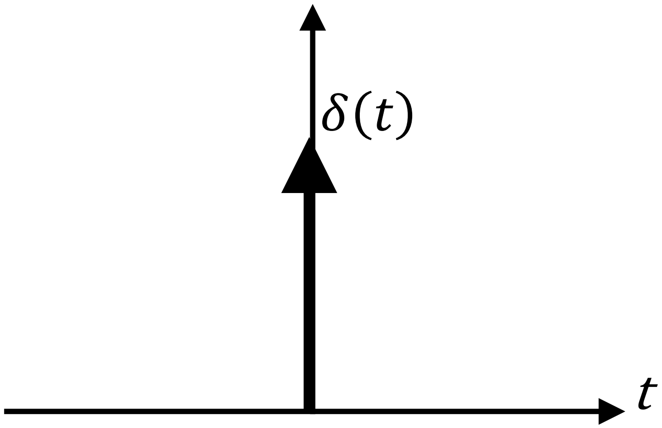
Slide by R A Peters

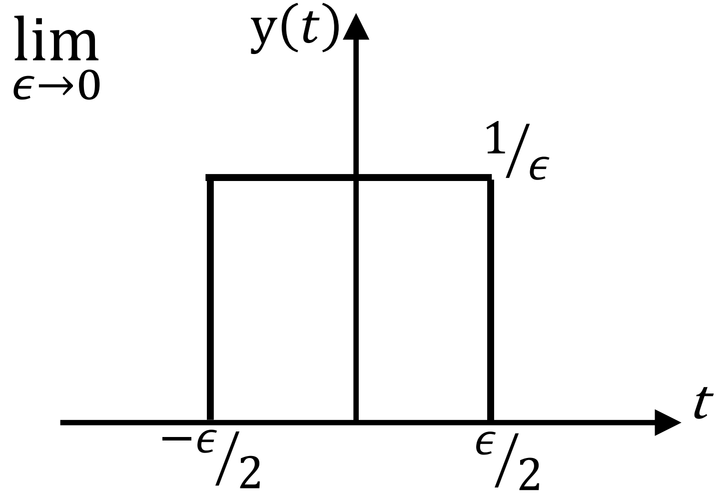
Digital Image Acquisition

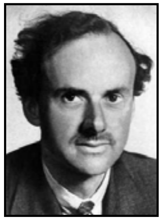


Slide by R A Peters

Modelling a Spatial Brightness Pulse - Dirac Delta-Function






Paul Dirac

Definition:

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

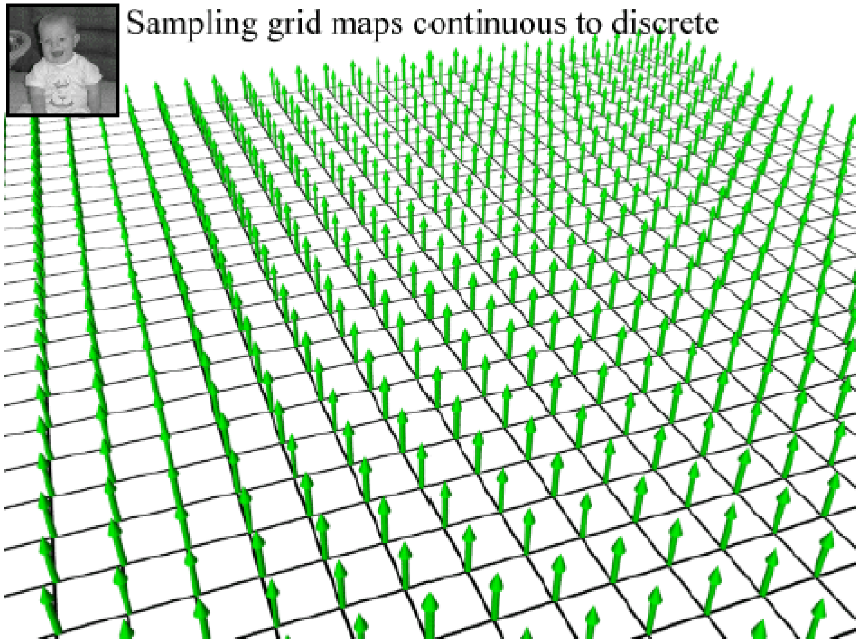
Intuitively:

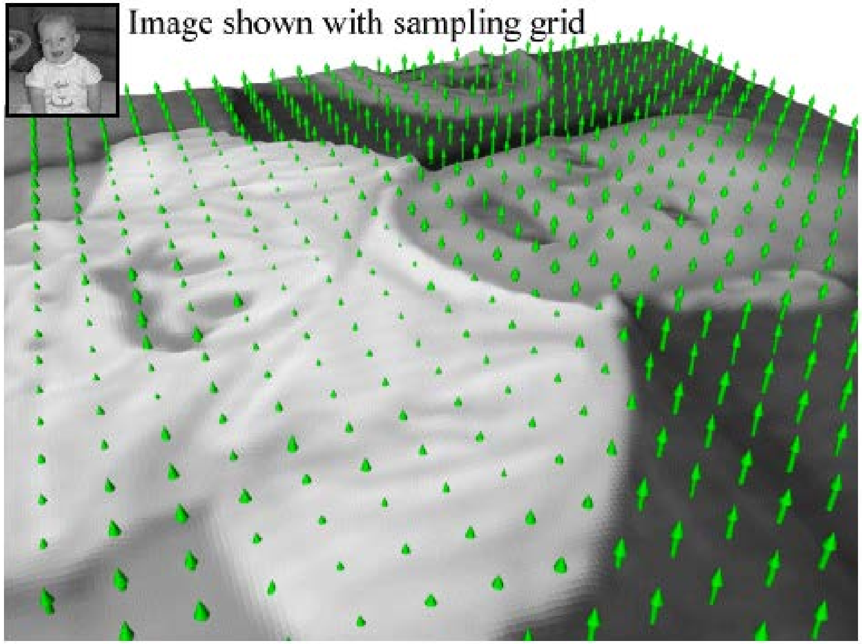
$$\delta(t) = \lim_{\epsilon \to 0}[y_\epsilon(t)]$$

Sifting Property:

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

Modelling a Spatial Brightness Pulse - Dirac Delta-Function





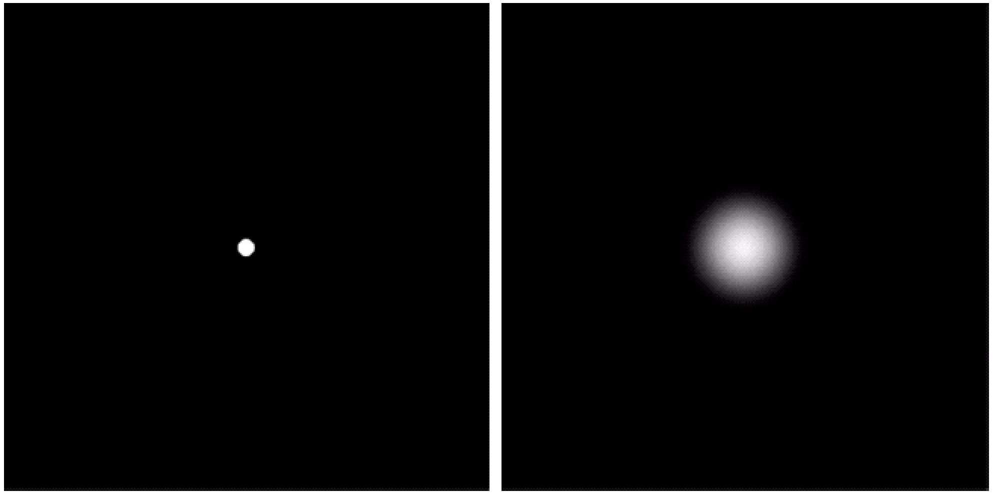
Images from Durand, Cutler and Hanrahan

The sifting property can be used to express a 2D 'image function' as a linear combination of 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

scene → Optical System → image



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

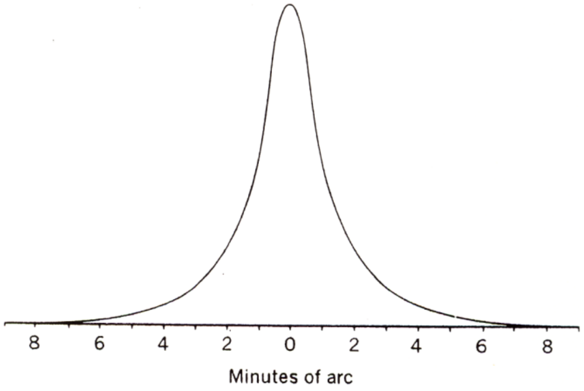
- However, optical systems are never ideal.

$\delta(x)$ → Optical System → $PSF(x)$

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

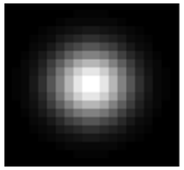


PSF Example

$$g(x,y) = f(x,y) * h(x,y)$$

$f(x,y)$ Original

$h(x,y)$



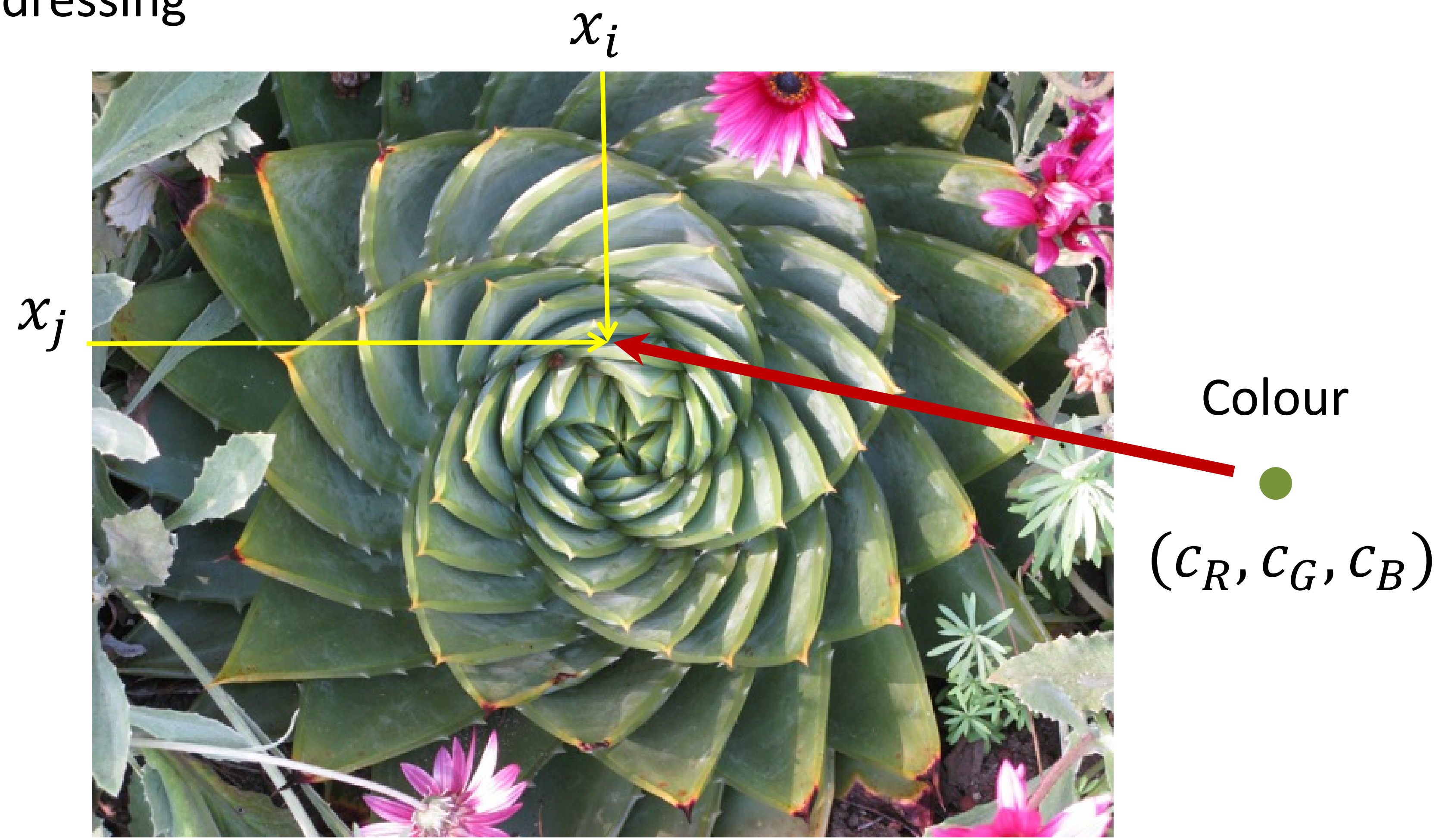
Blurry outcome $g(x,y)$

$h(x,y)$ is the PSF of the imaging device.

Adapted from a slide by A. Zisserman

How to model an image?

Addressing



How to represent a digital image?

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

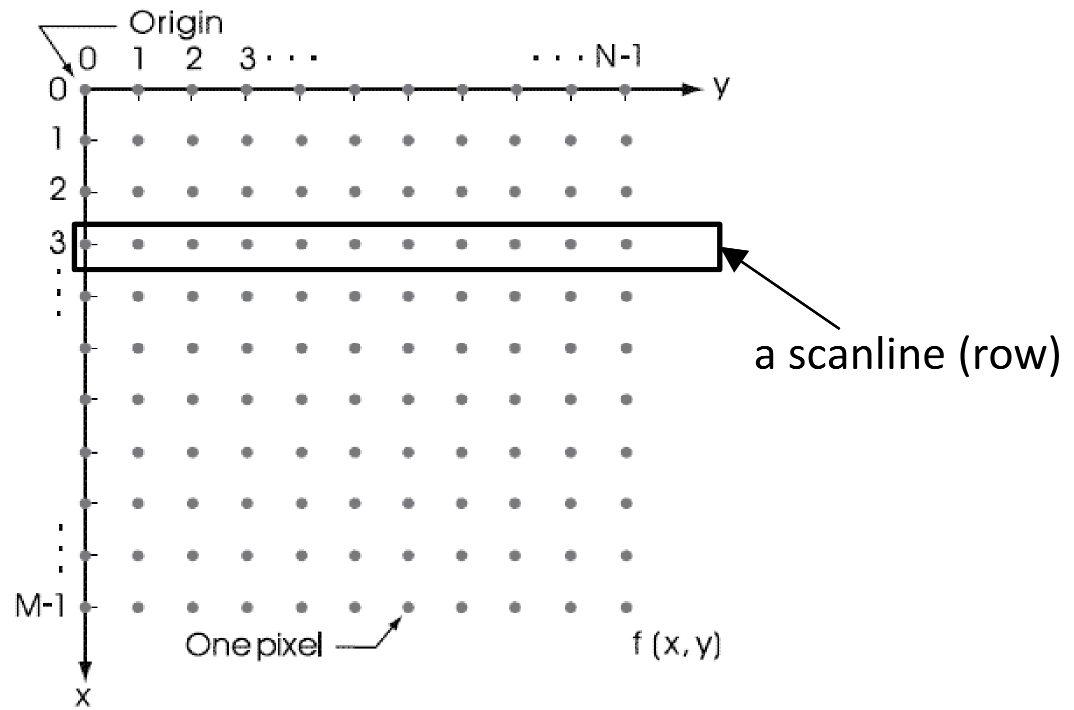
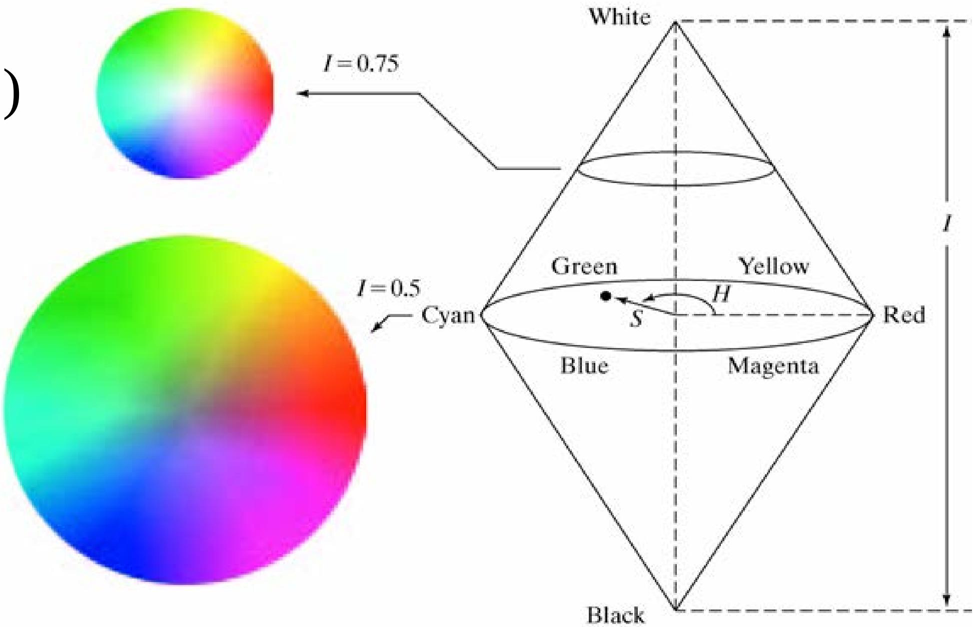
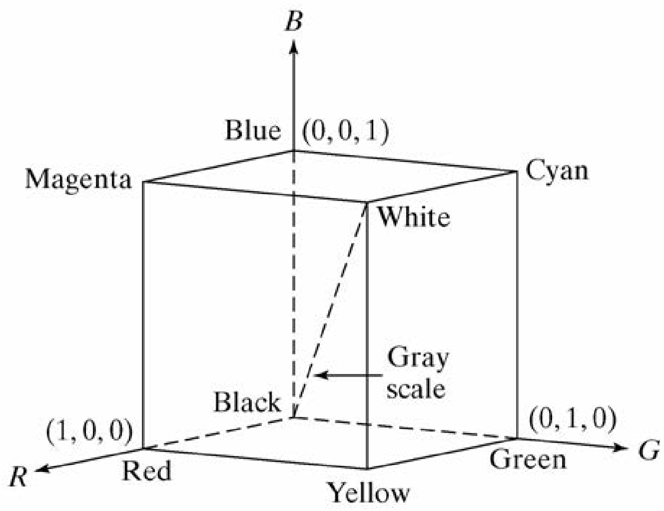


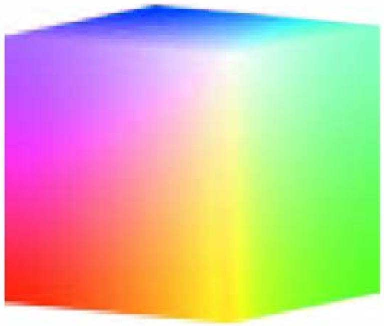
Image Representation: Colour Spaces

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



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





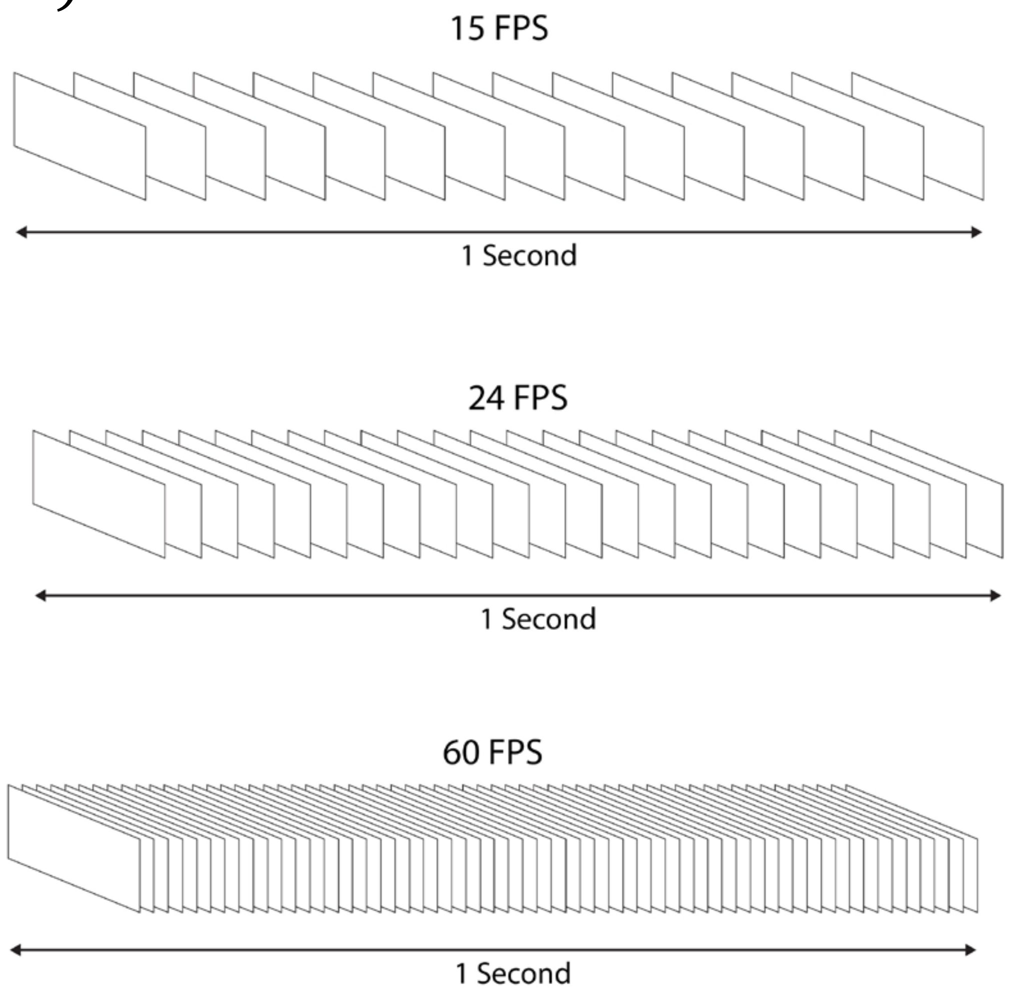
Other examples are:

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

...

How to represent a video?

$$f[x, y, t] = (R, G, B)$$



From quora.com

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

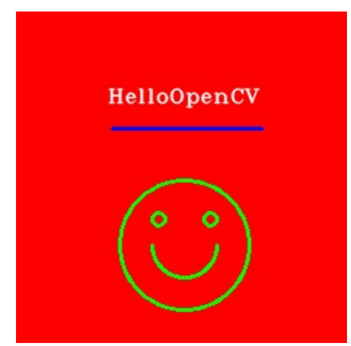
3 channels

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;
}
```

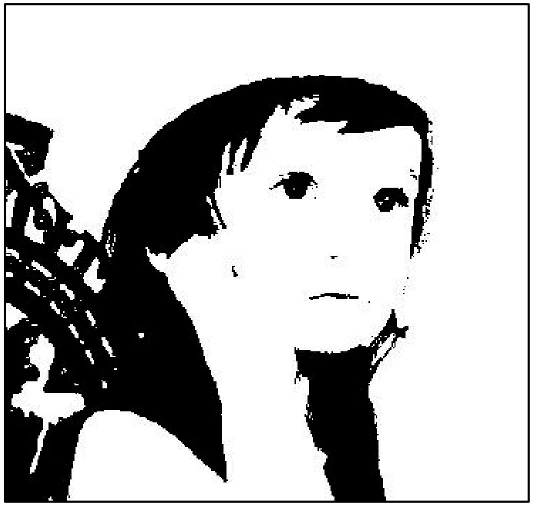


Quantization of the Image Function

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

16 levels

6 levels

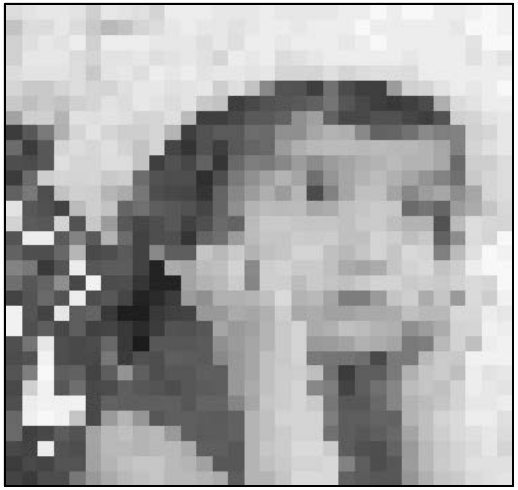


2 levels

The effect of very sparse sampling … is often ALIASING

256 x256

64x64



32x32

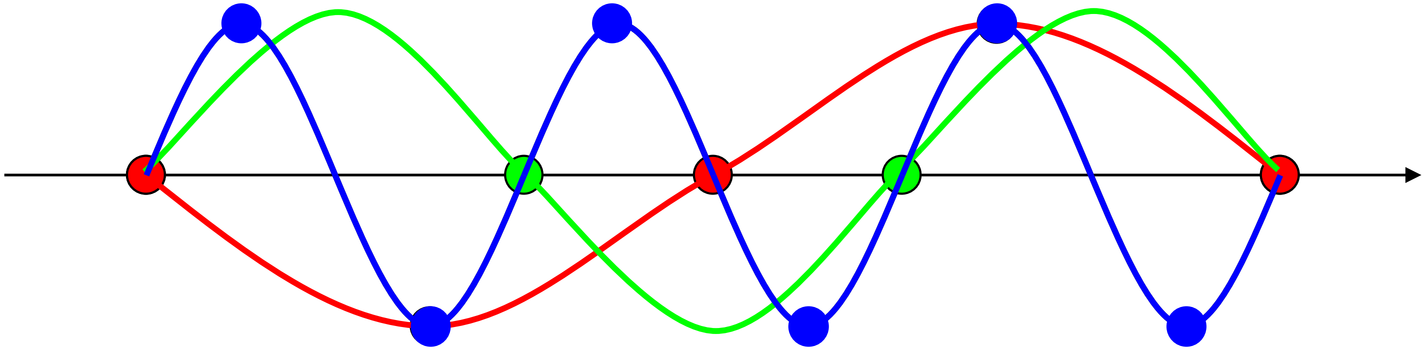
Anti-aliasing can be achieved by removing all spatial frequencies above a critical limit (so-called Shannon-Nyquist Limit).

Shannon's Sampling Theorem

Claude Shannon

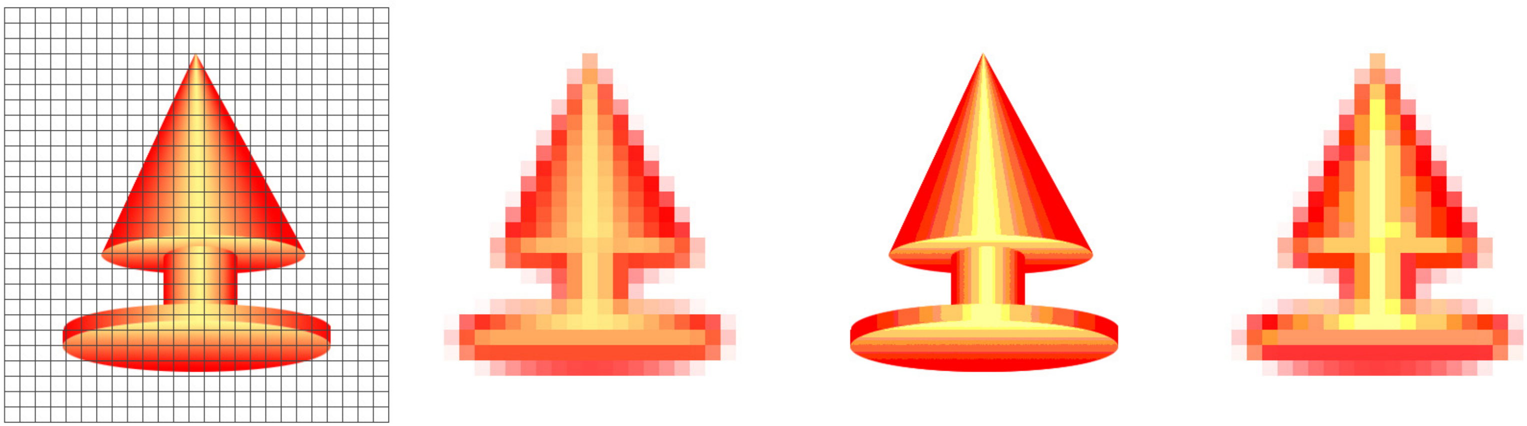
"An analogue signal containing components up to some maximum frequency ***u*** may be completely reconstructed by regularly spread samples, provided the sampling rate is above 2***u*** samples per second."

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



Sampling and Quantization Summary

pixel grid



real image

sampled

quantized

sampled & quantized

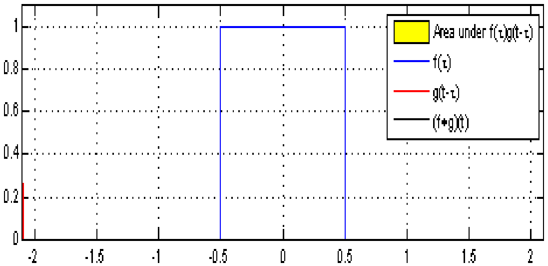
Slide by R A Peters

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***

f(x)

2D Discrete Convolution

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

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

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

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

f ***** ***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:

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

$$\begin{array}{c|ccc|}
 & -1 & 0 & 1 \\
\hline
-1 & -1 & 0 & 1 \\
0 & -2 & 0 & 2 \\
1 & -1 & 0 & 1 \\
\hline
\end{array}
\qquad
\begin{array}{c|c|c|c|c|c|}
 & & y-1 & y & y+1 & \\
\hline
 & & & & & \\
\hline
x-1 & & 43 & 12 & 61 & \\
\hline
x & & 44 & \boxed{45} & 60 & \\
\hline
x+1 & & 43 & 50 & 61 & \\
\hline
 & & & & & \\
\hline
\end{array}$$

$\rightarrow$ **68**

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

1	2	1

Spatial Low/High Pass Filtering

Low Pass

***h*=**

1	1	1
1	1	1
1	1	1

f

f * h

High Pass

***h*=**

-1	-1	-1
-1	8	-1
-1	-1	-1