COMP9517

Computer Vision

2022 Term 2 Week 1

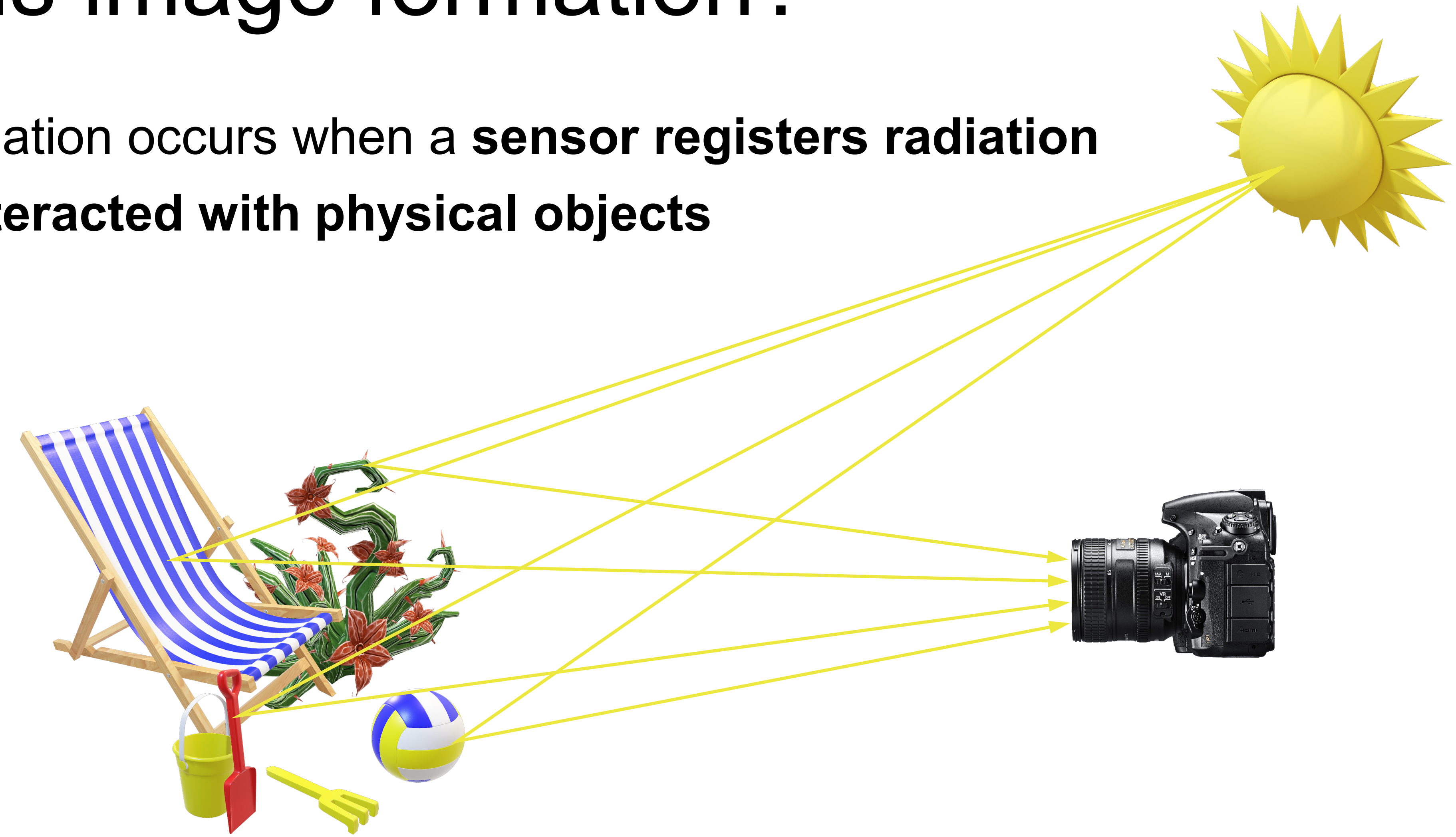
Professor Erik Meijering

UNSW SYDNEY

Image Formation

What is image formation?

Image formation occurs when a **sensor registers radiation** that has **interacted with physical objects**



Geometry of image formation

Mapping world coordinates (3D) to image coordinates (2D)

- Pinhole camera model
- Projective geometry
- Projection matrix

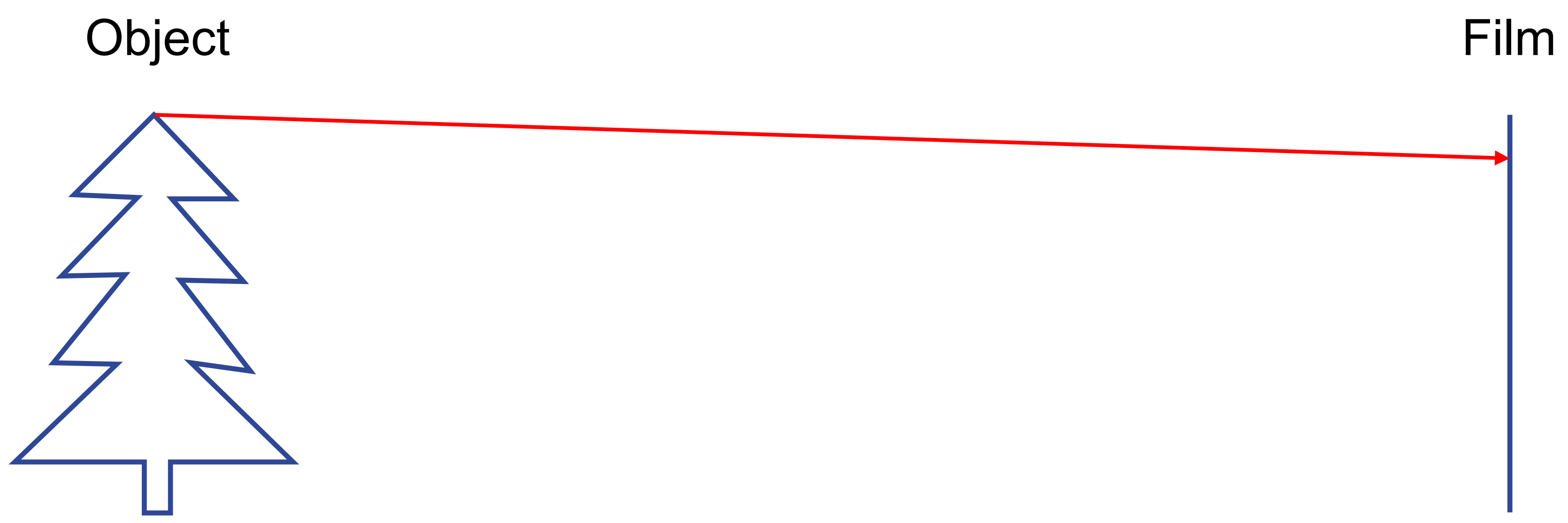
Image formation

Idea 1: Put a piece of film in front of an object

Do we get a reasonable image?

Image formation

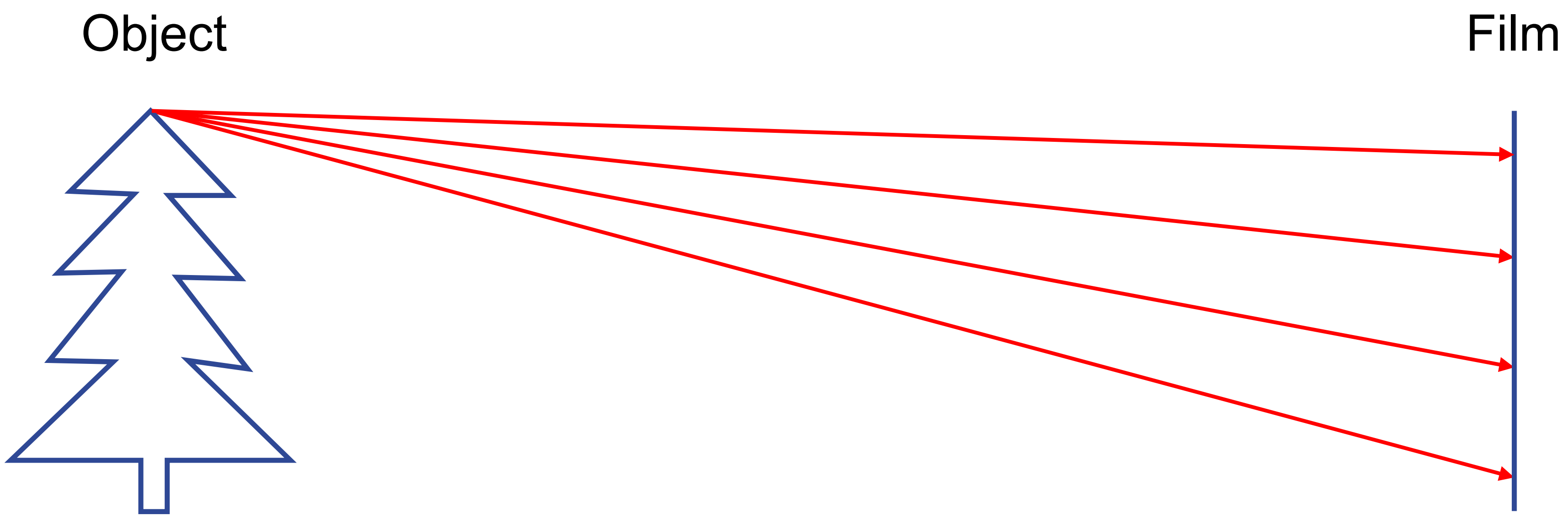
Idea 1: Put a piece of film in front of an object



Do we get a reasonable image?

Image formation

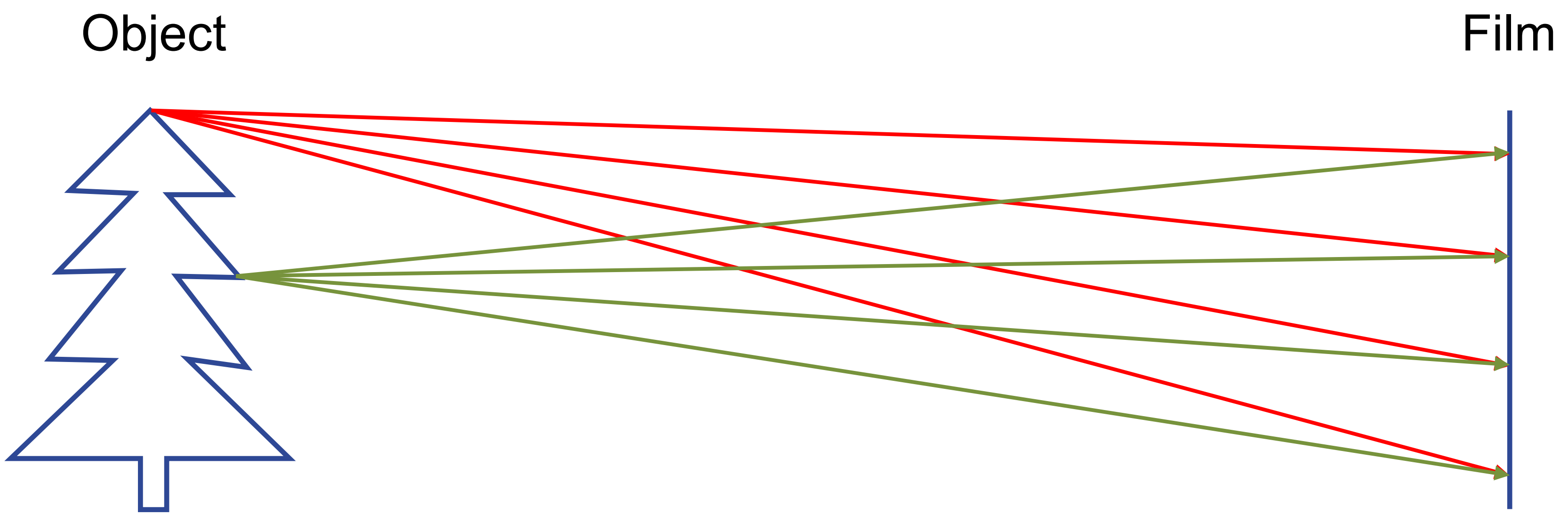
Idea 1: Put a piece of film in front of an object



Do we get a reasonable image?

Image formation

Idea 1: Put a piece of film in front of an object

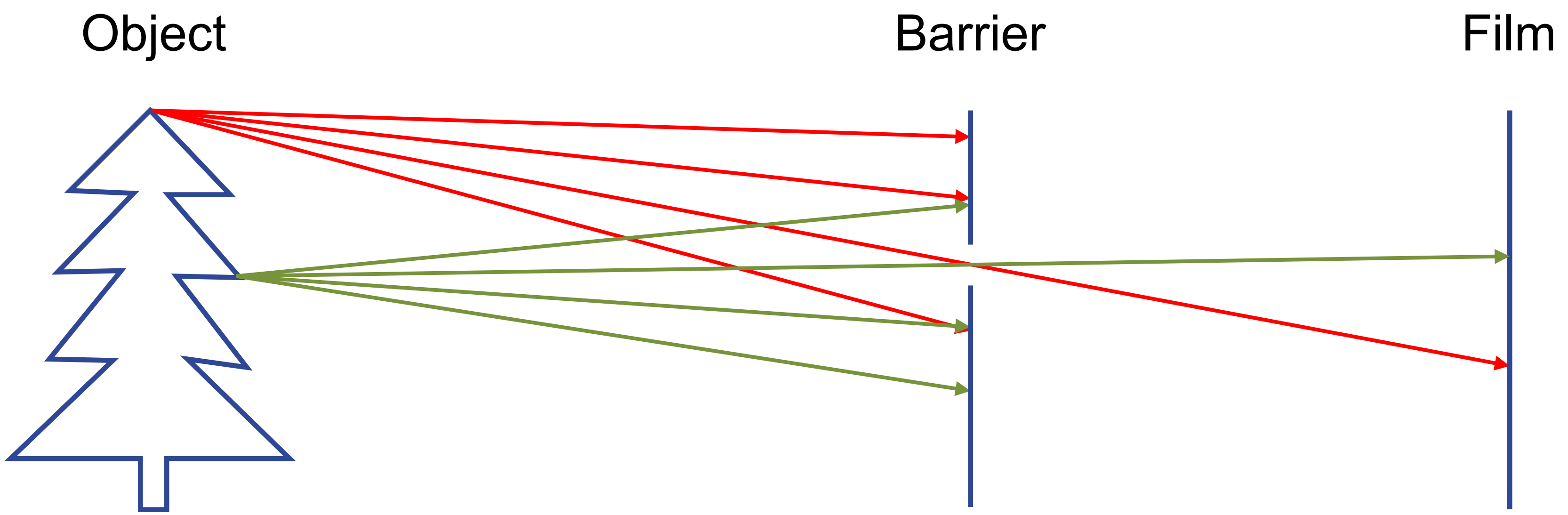


All object points are projected to all points on the film

The resulting image is completely blurred

Image formation

Idea 2: Add a barrier to block off most of the rays

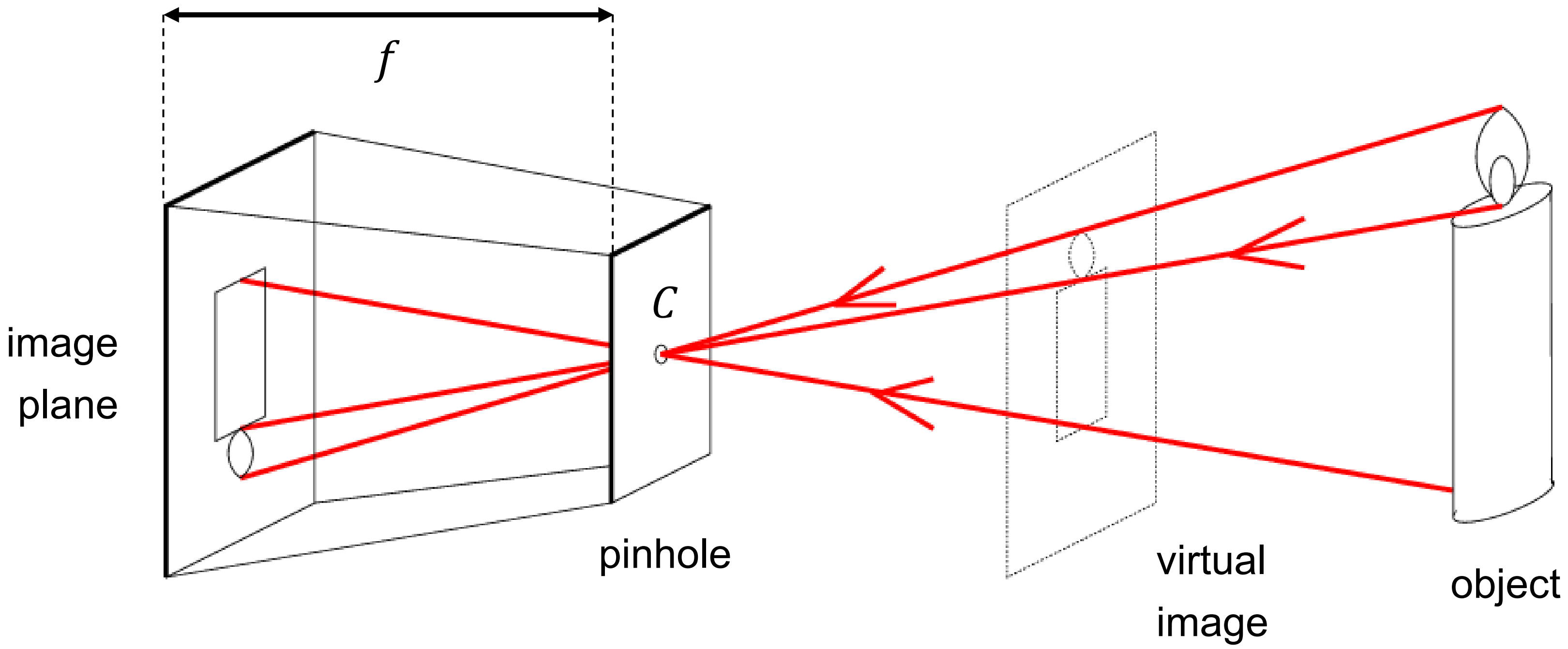


This reduces blurring

Opening known as the **pinhole** or **aperture**

Object points are projected to unique points on the film

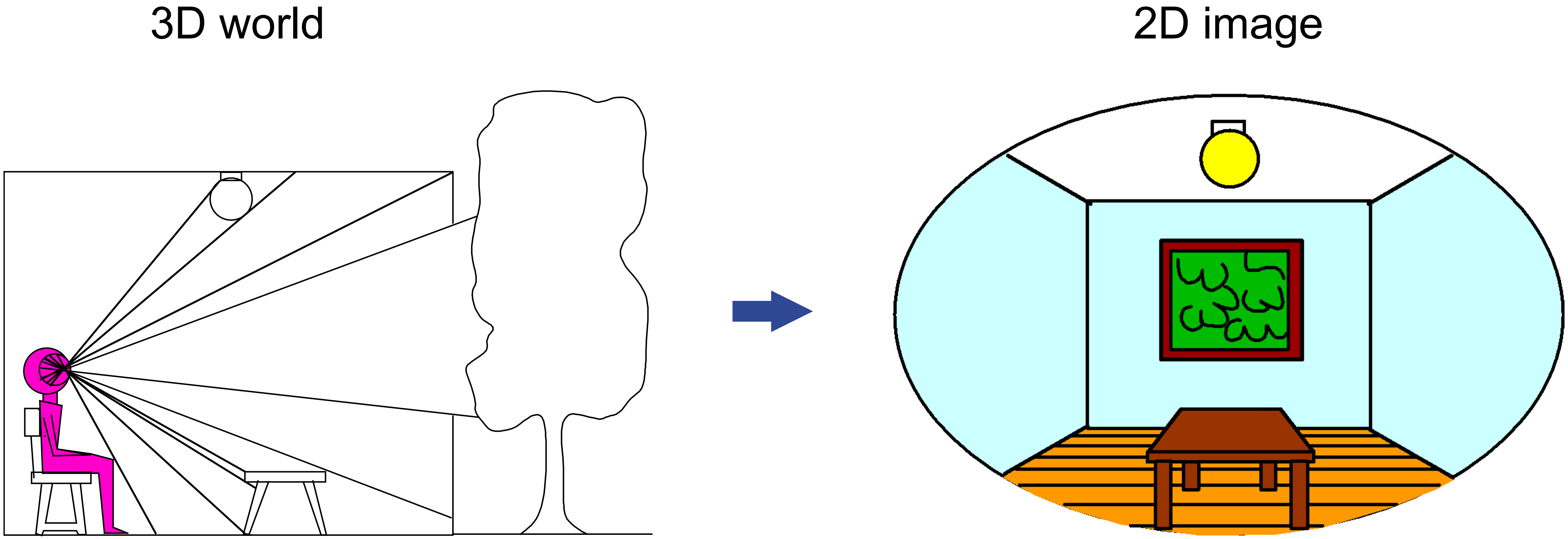
Pinhole camera model



f = focal length
C = camera centre

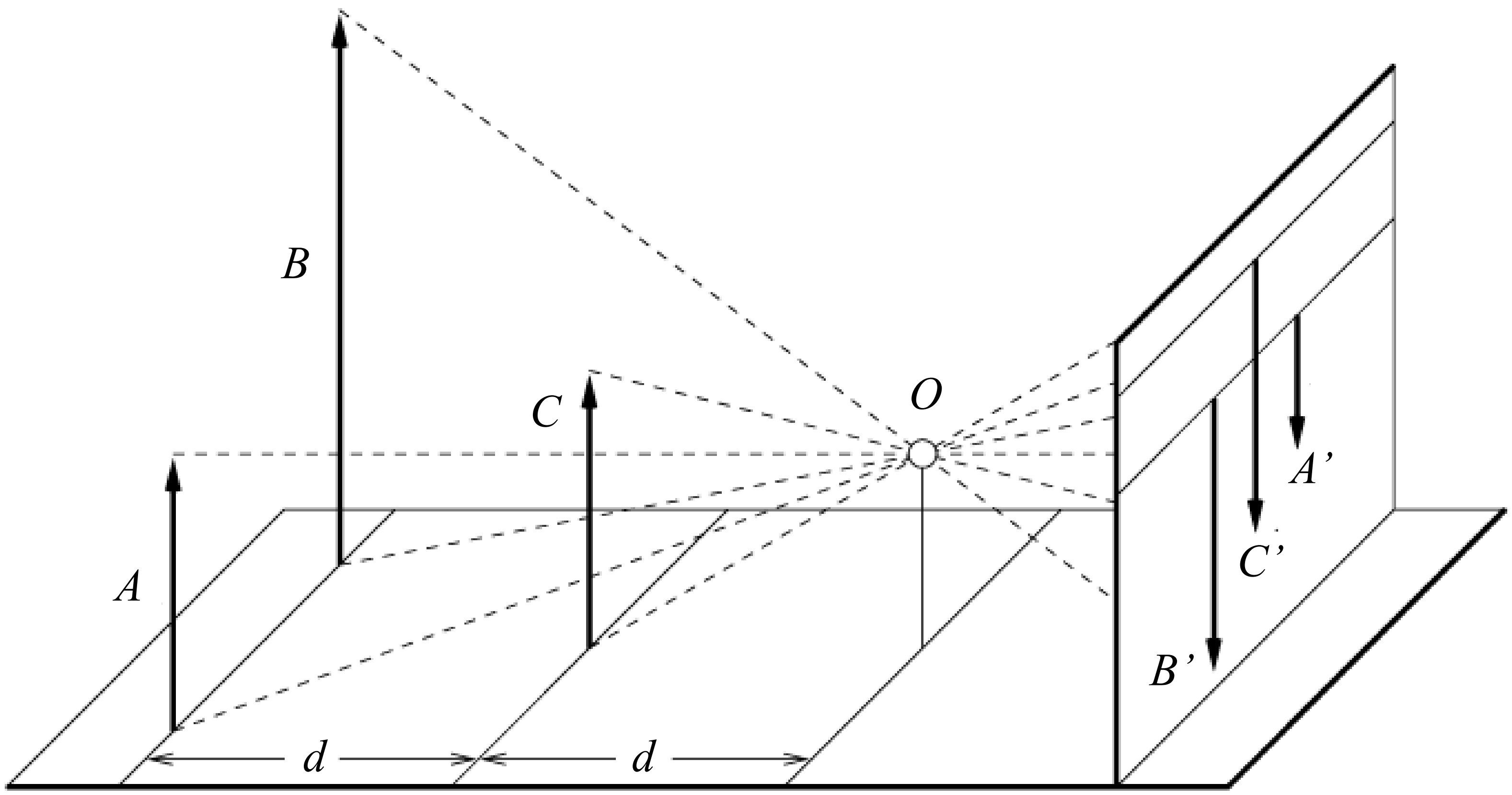
Figure from Forsyth

Dimensionality reduction machine



Projection can be tricky…

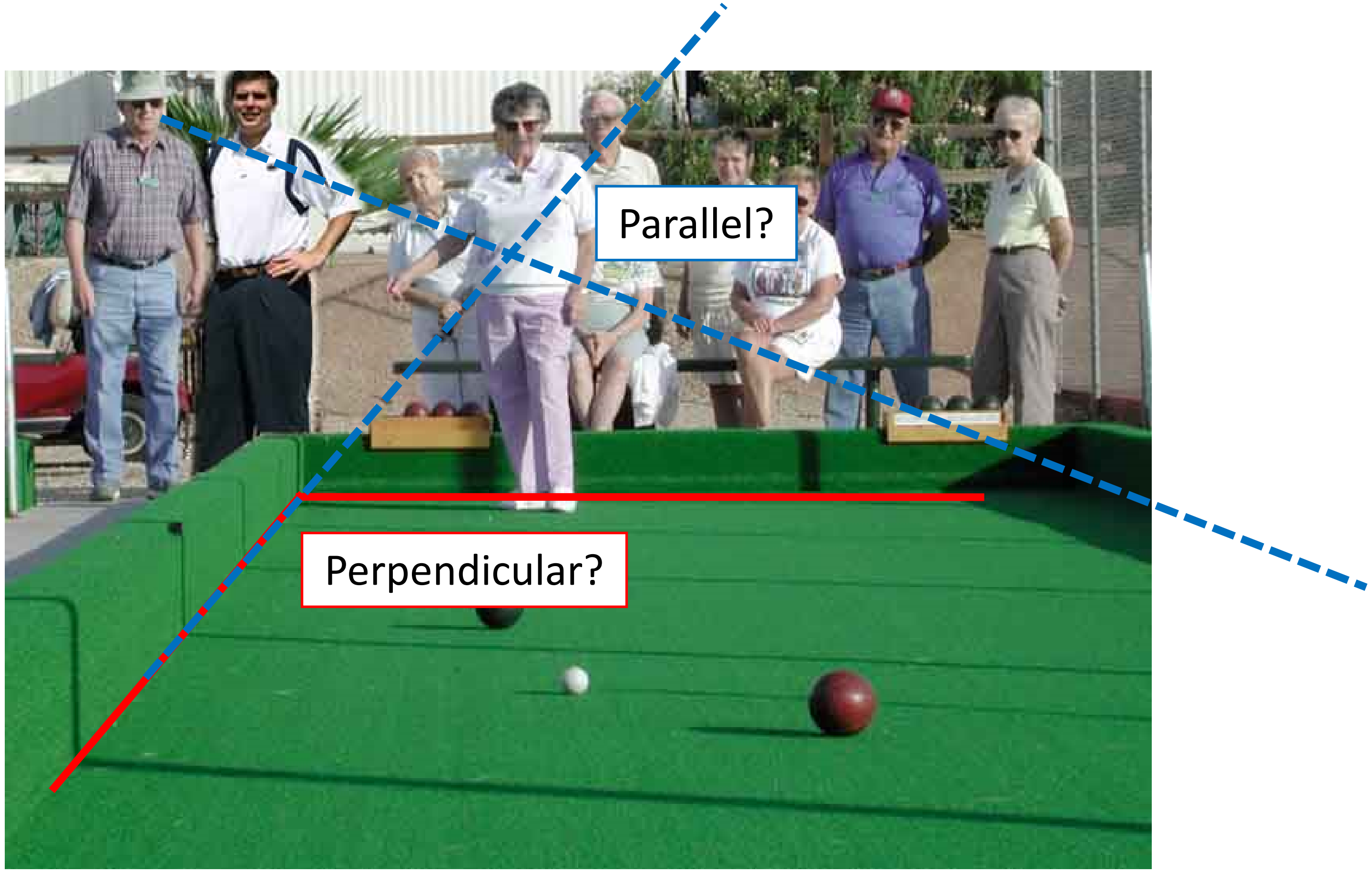
Projective geometry



Length and area are not preserved

Figure from Forsyth

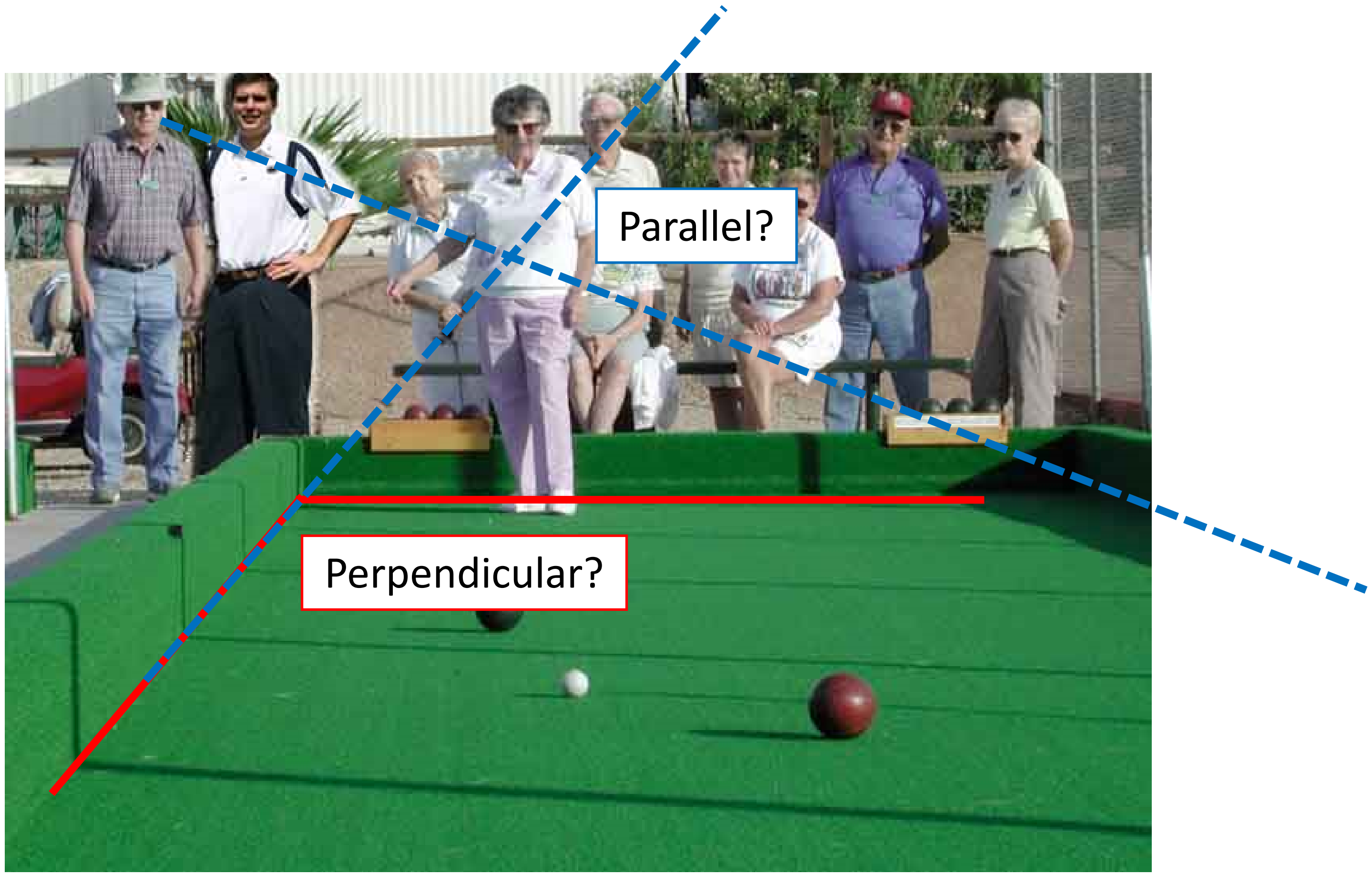
Projective geometry



What is lost?

Length and angles are not preserved

Projective geometry



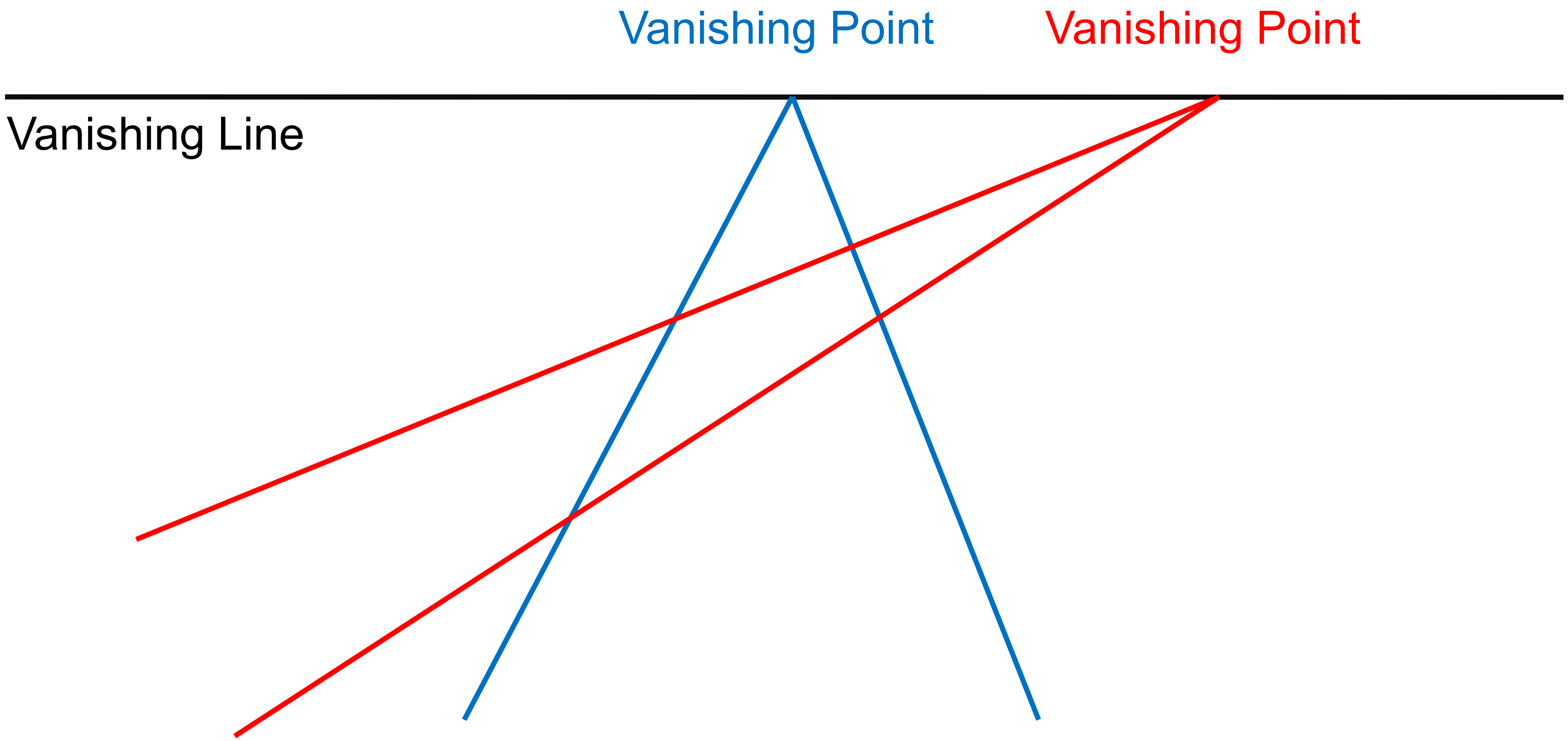
What is preserved?

Straight lines are still straight

Vanishing points and lines

Parallel lines in the 3D world
intersect in the 2D image
at a “vanishing point”

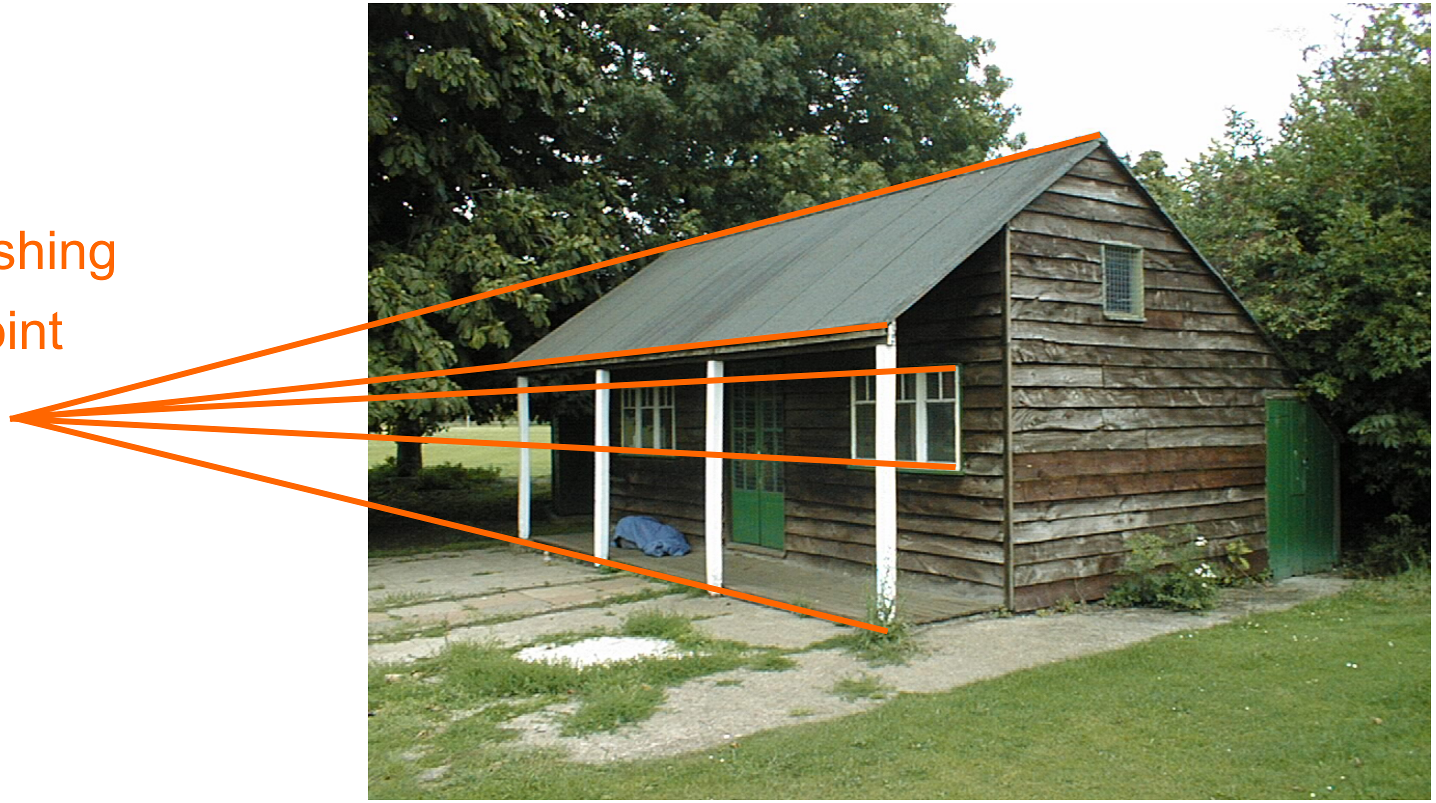
Vanishing points and lines



Vanishing points and lines

Photo from Criminisi

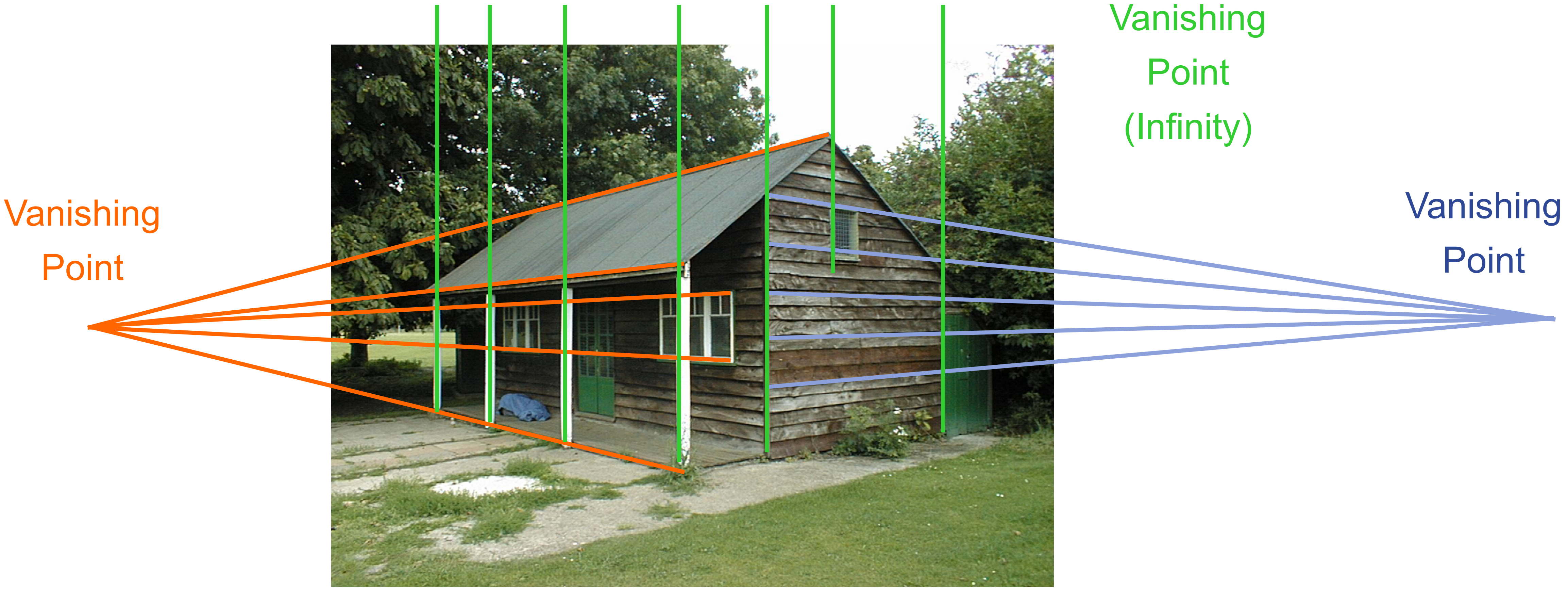
Vanishing points and lines



Vanishing Point

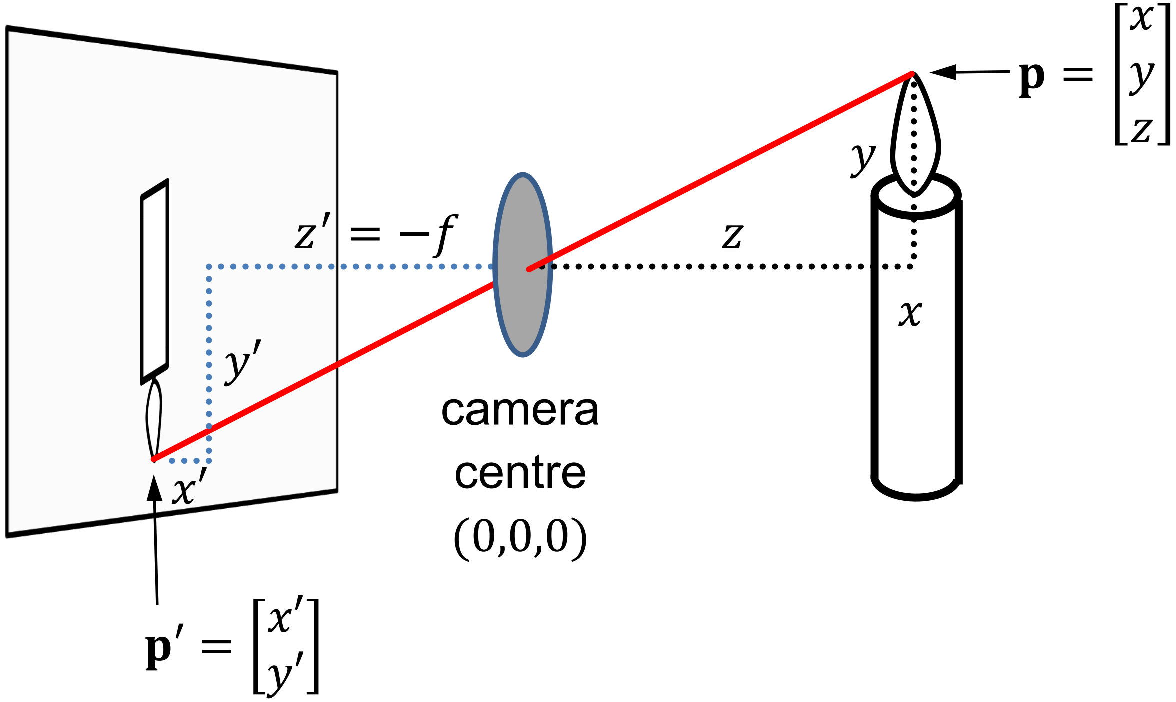
Vanishing points and lines

Vanishing points and lines



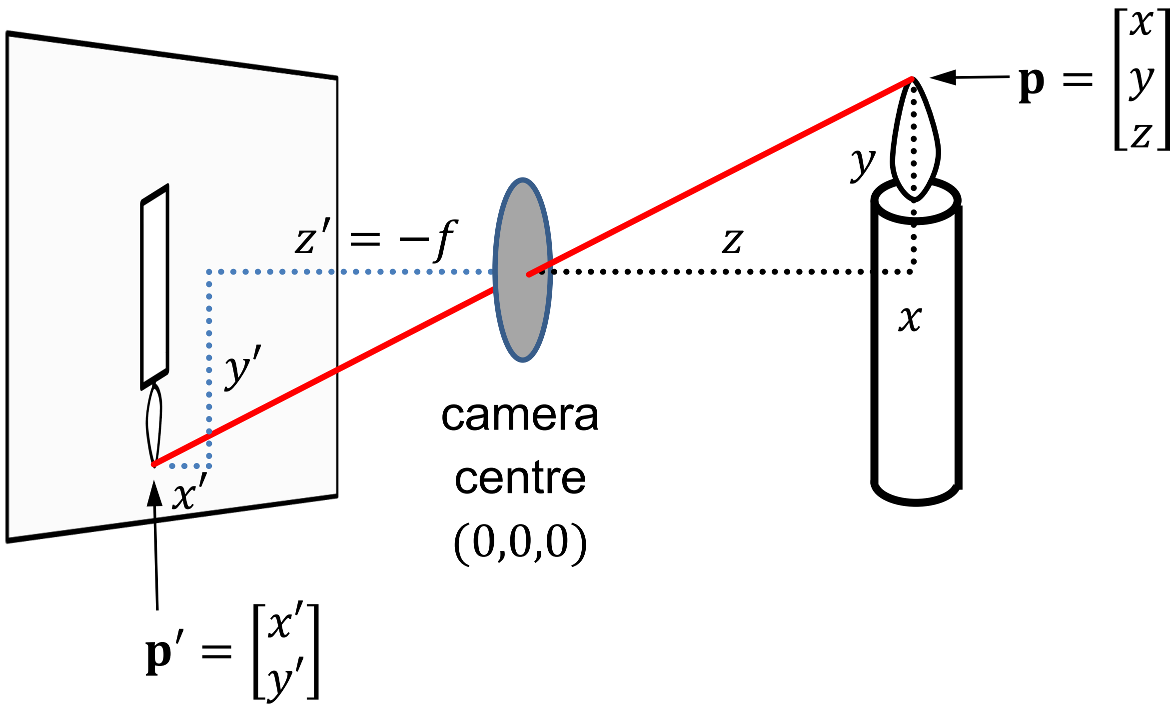
Projection mathematics

From **world coordinates** to **image coordinates**



Projection mathematics

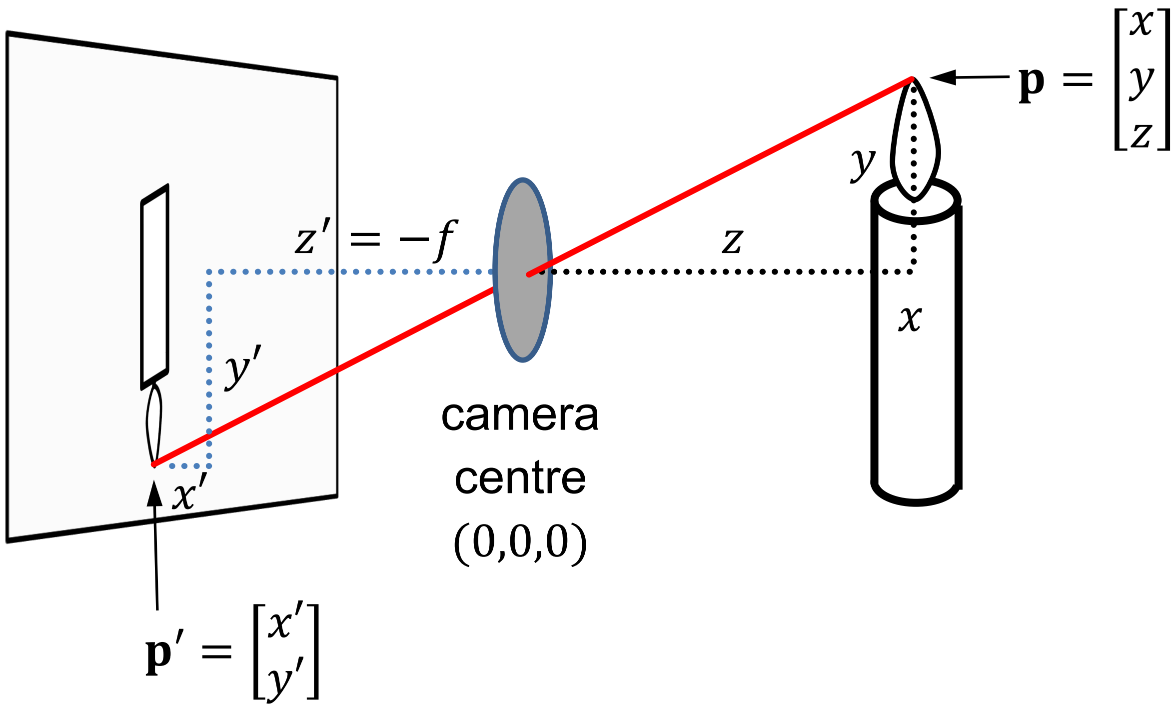
From **world coordinates** to **image coordinates**



If $x = 2, y = 3, z = 5$, and $f = 2$,
what are x' and y' ?

Projection mathematics

From **world coordinates** to **image coordinates**



If $x = 2, y = 3, z = 5$, and $f = 2$, what are x' and y' ?

$$x' = -x \cdot \frac{f}{z} \qquad \Rightarrow \qquad x' = -2 \cdot \frac{2}{5}$$

$$y' = -y \cdot \frac{f}{z} \qquad \qquad y' = -3 \cdot \frac{2}{5}$$

Perspective projection

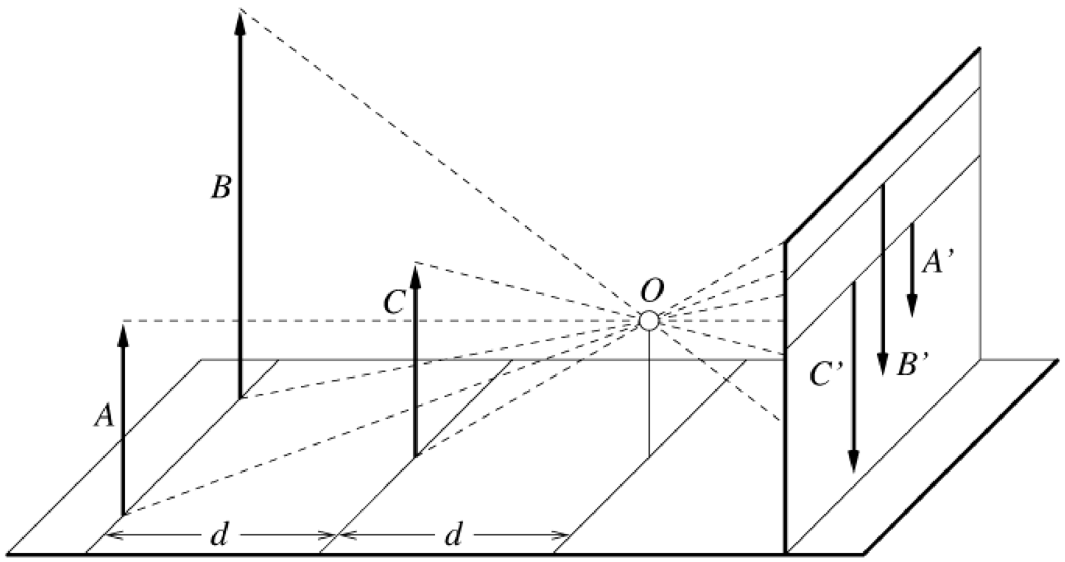
- Apparent size of object depends on its distance: far objects appear smaller

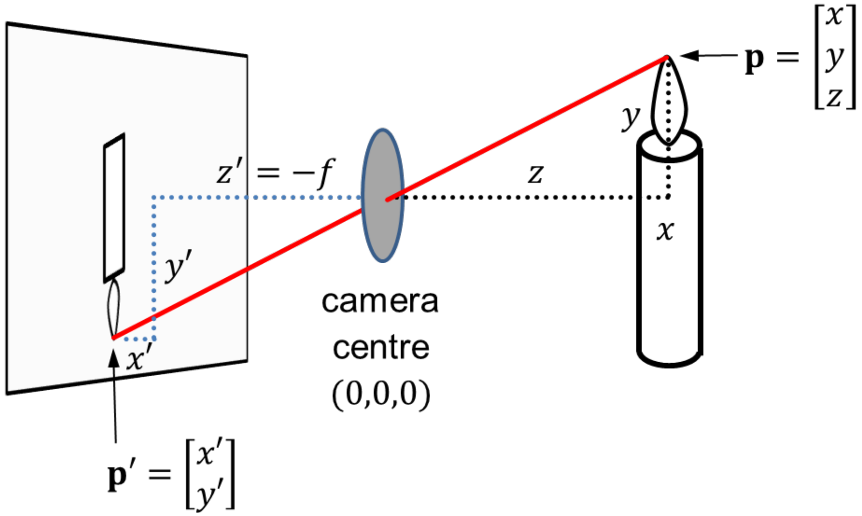
- By similar triangles:

$$(x', y', z') = (-f\frac{x}{z}, -f\frac{y}{z}, -f)$$

- Ignore third coordinate and mirror:

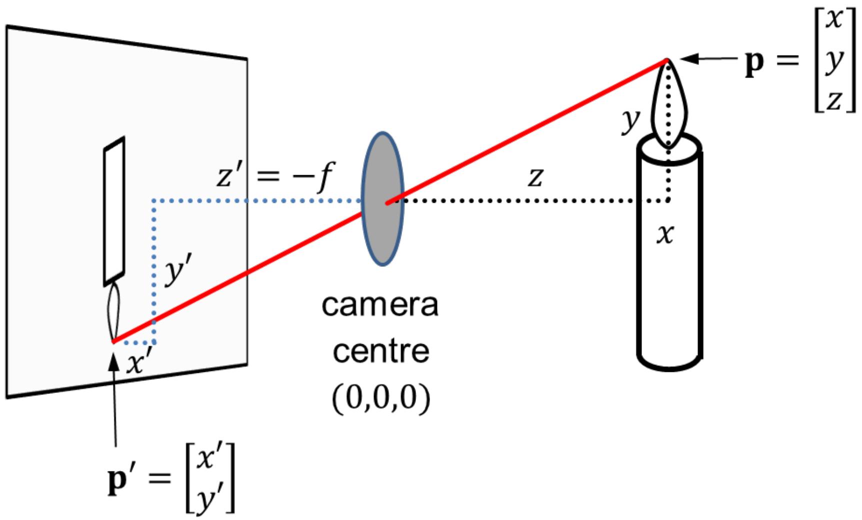
$$(x', y') = (f\frac{x}{z}, f\frac{y}{z})$$



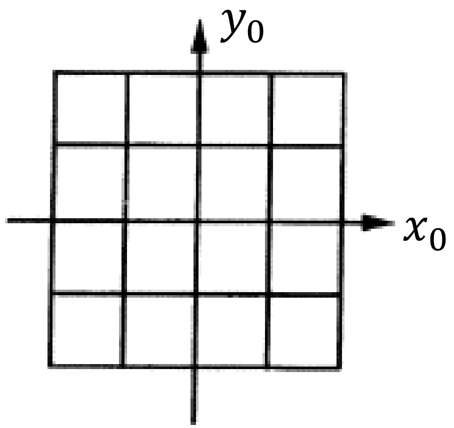


Affine projection

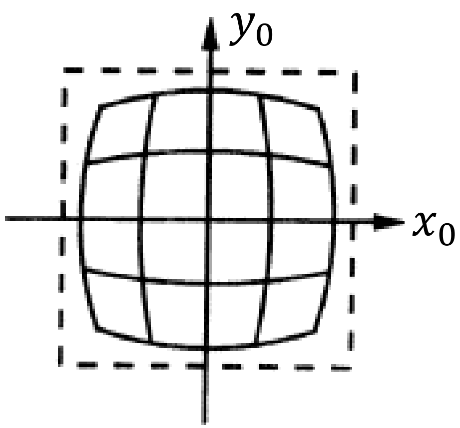
- Suitable when scene depth is small relative to average distance from camera
- Let magnification $m = f/z_0$ be a positive constant and all points in the scene have approximately constant distance z_0 to the camera
- Leads to weak perspective projection:
  $(x', y') = (mx, my)$
- Orthographic projection when $m = 1$:
  $(x', y') = (x, y)$



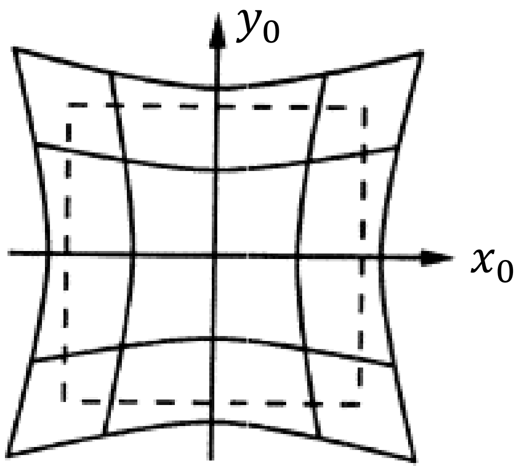
Beyond pinholes: radial distortions



No distortion



Barrel distortion



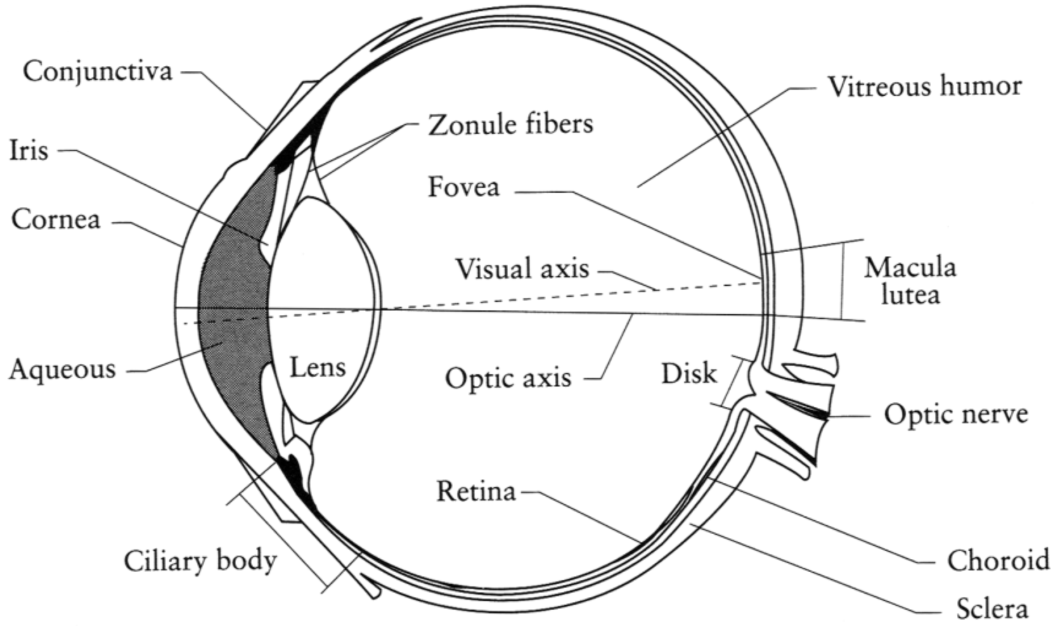
Pincushion distortion

Barrel distortion corrected

Image from Martin Habbecke

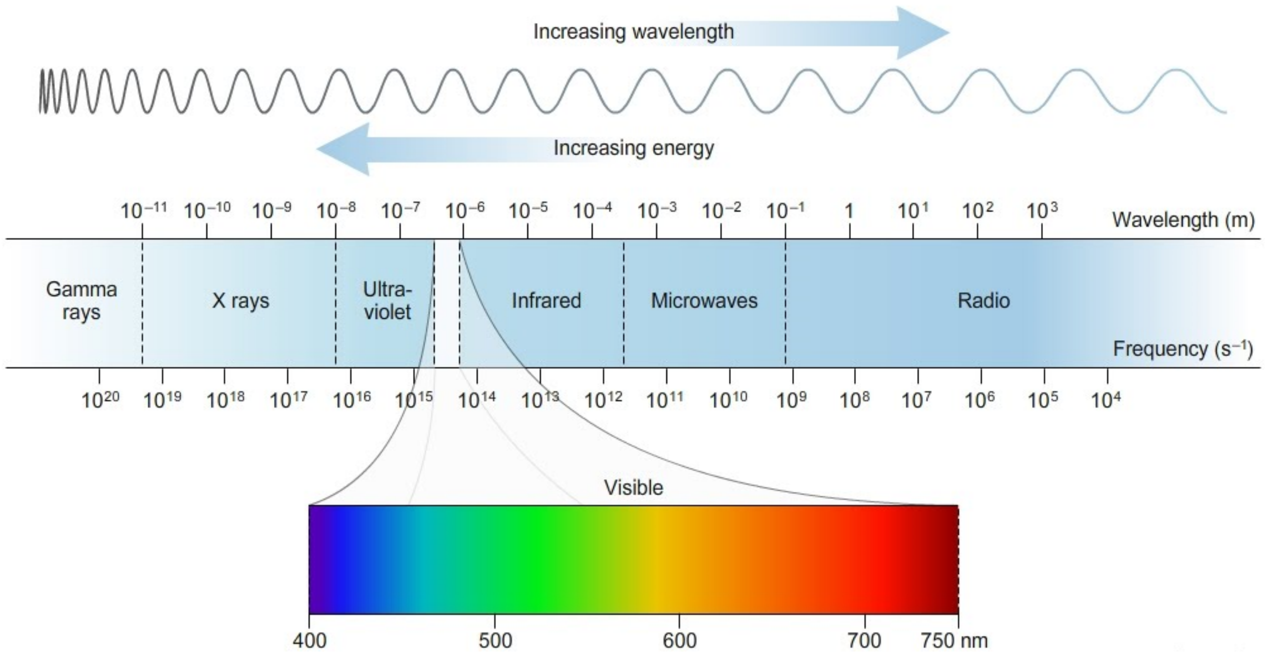
Comparing with human vision

- Cameras imitate the frequency response of the human eye so it is good to know something about it
- Computer vision probably would not get as much attention if biological vision (especially human vision) had not proven that it is possible to make important judgements from 2D images

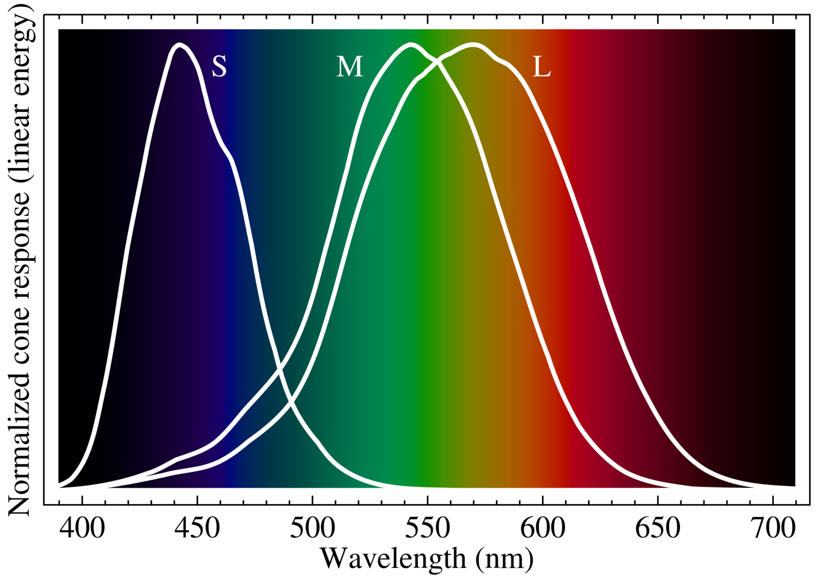


The Eye

Electromagnetic spectrum

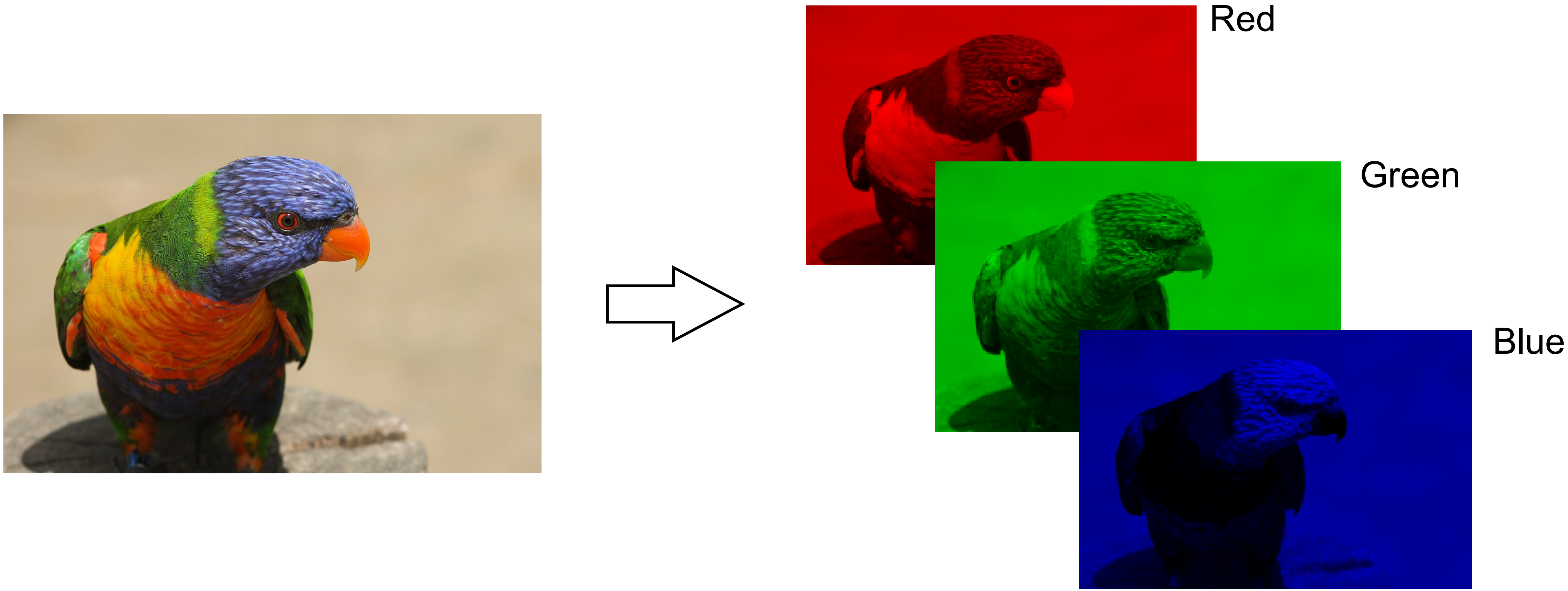


https://sites.google.com/site/chempendix/em-spectrum



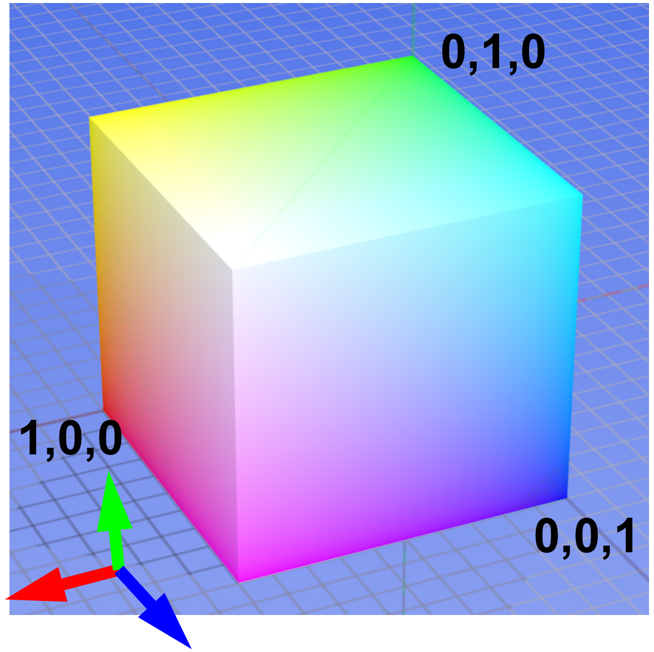
Normalized responsivity spectra of human cone cells (S, M, L types)

Colour represented by RGB images



Colour spaces: RGB

Default colour space in vision



Drawback: strongly correlated channels

R

(G=0,B=0)

G

(R=0,B=0)

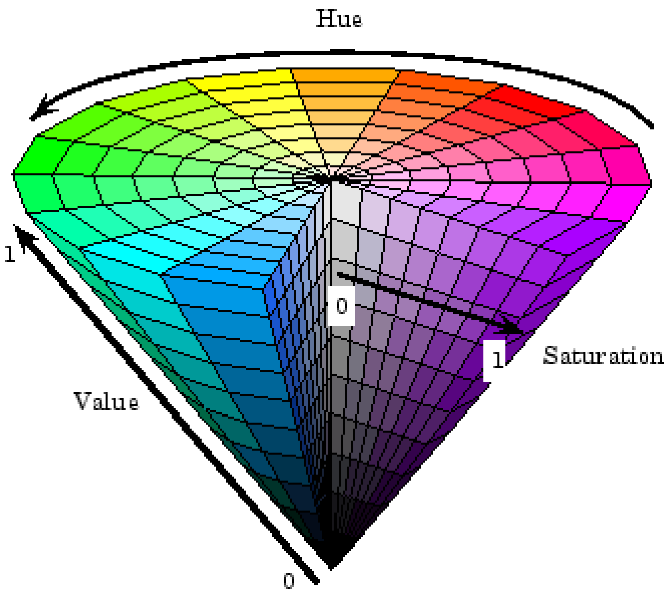
B

(R=0,G=0)

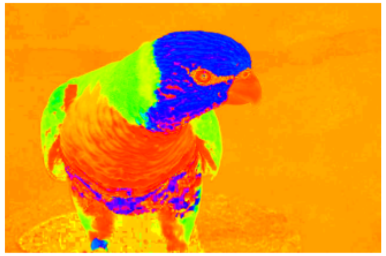
http://en.wikipedia.org/wiki/File:RGB_color_solid_cube.png

Colour spaces: HSV

Intuitive colour space



Drawback: confounded channels



H

(S=1,V=1)

S

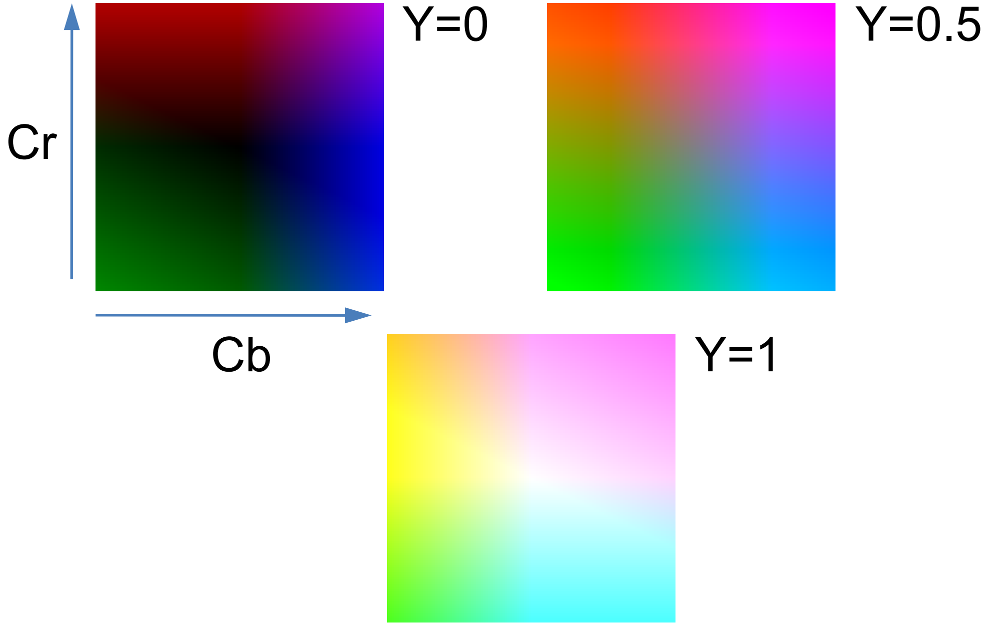
(H=1,V=1)

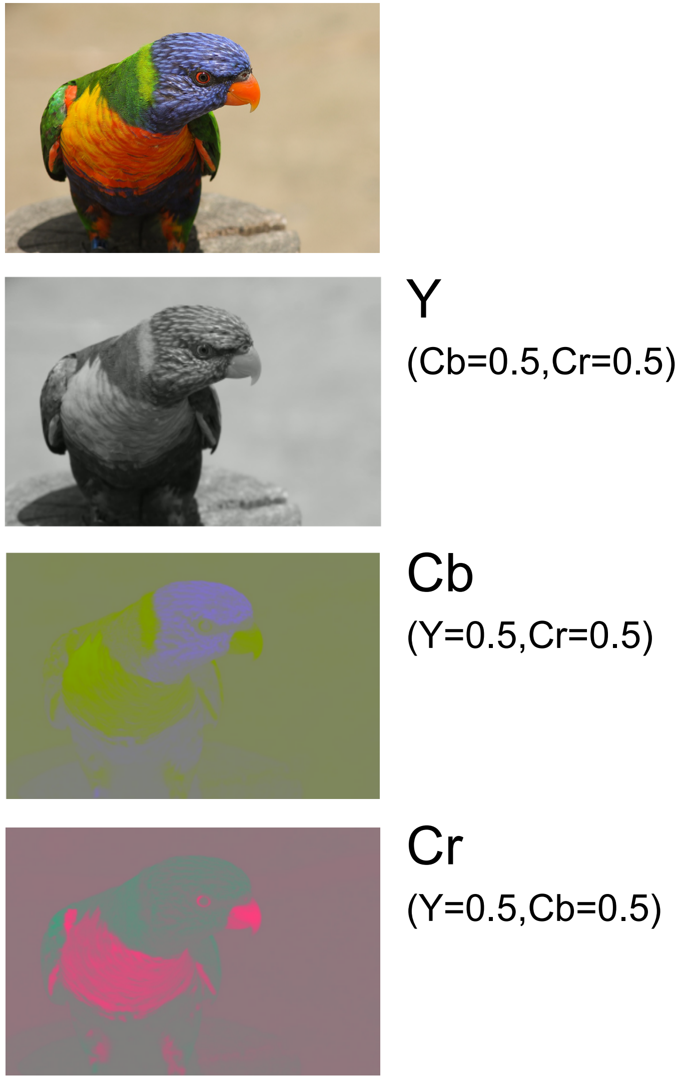
V

(H=1,S=0)

Colour spaces: YCbCr

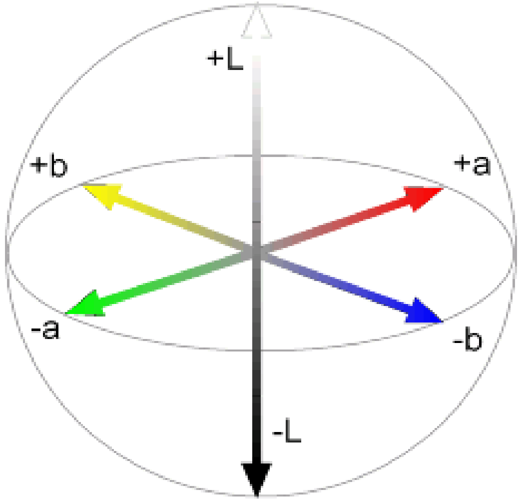
Fast to compute, good for compression, used by TV





Colour spaces: L*a*b*

"Perceptually uniform" colour space



a.k.a. CIELAB

Any numerical change corresponds to similar perceived change in color: Euclidean distances make sense

L
(a=0, b=0)

a
(L=65, b=0)

b
(L=65, a=0)

Digital image formation

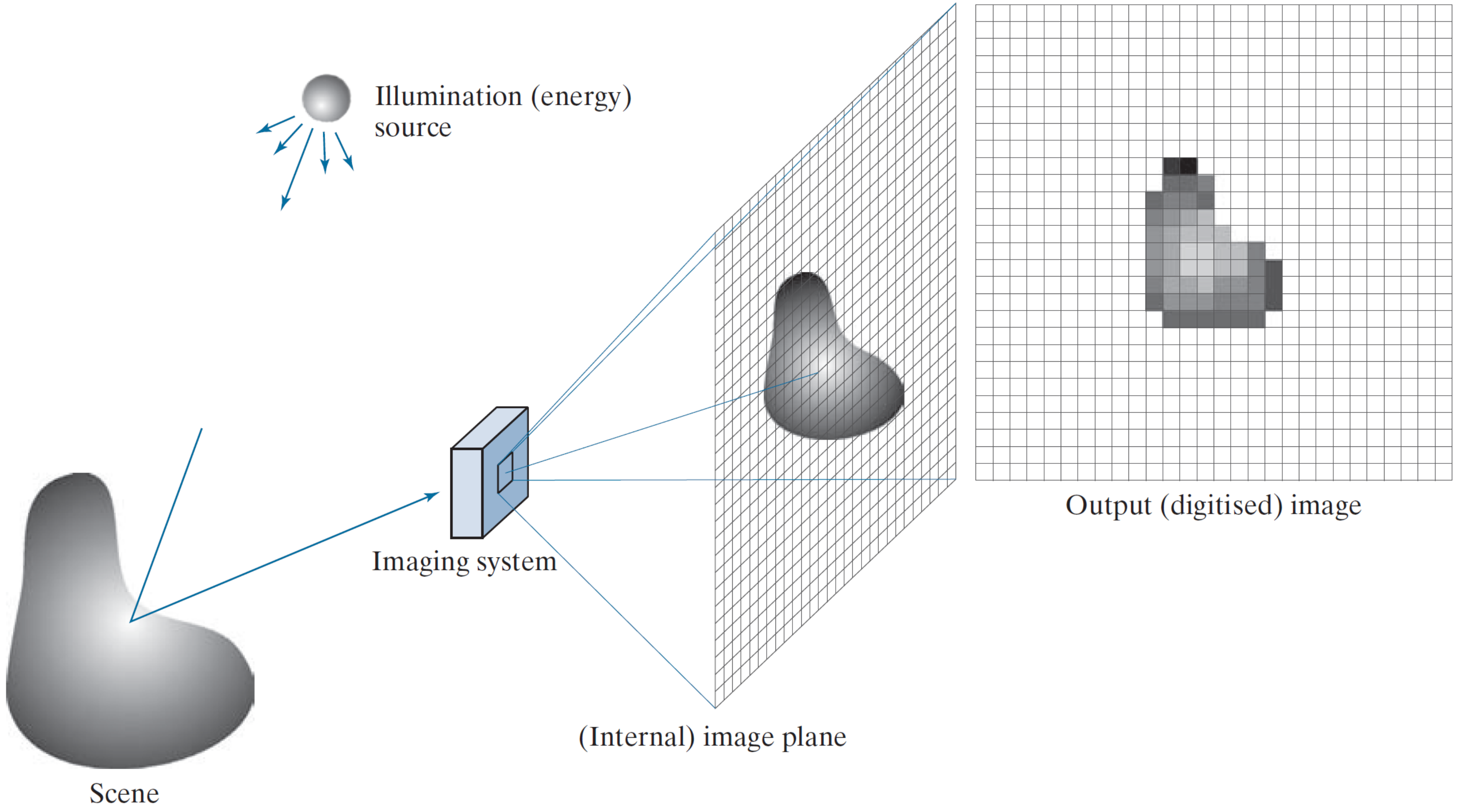
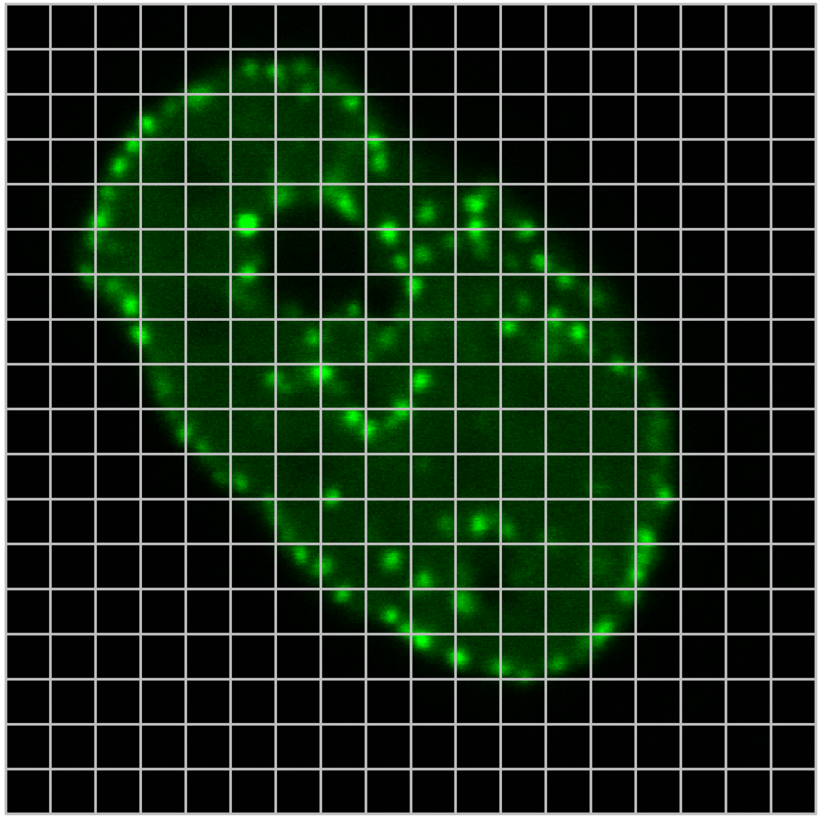


Image from Gonzalez & Woods 2018

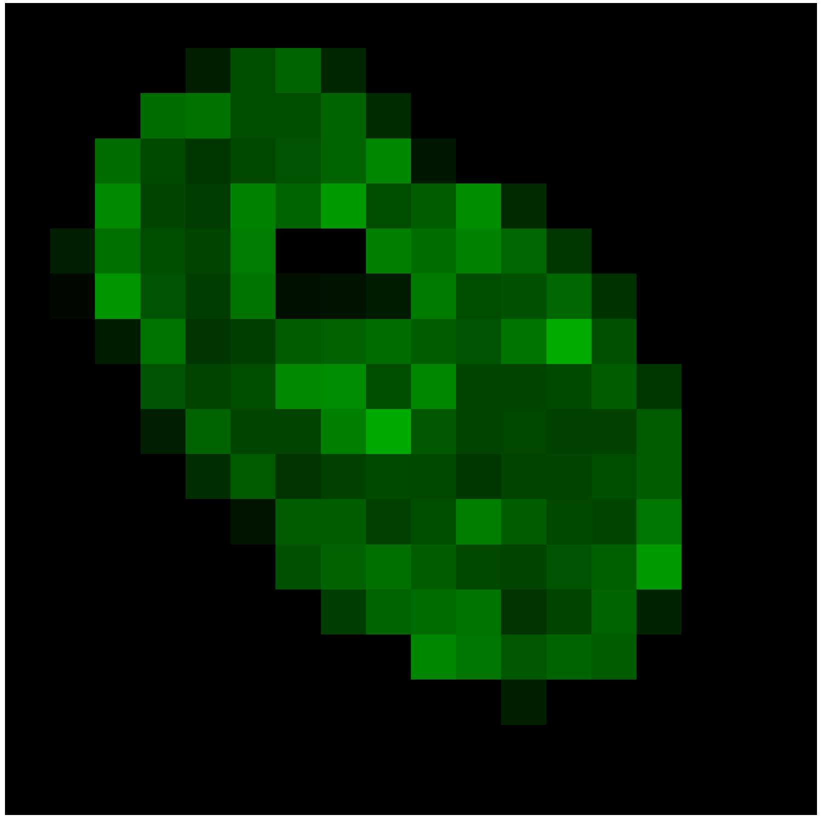
Digital image formation



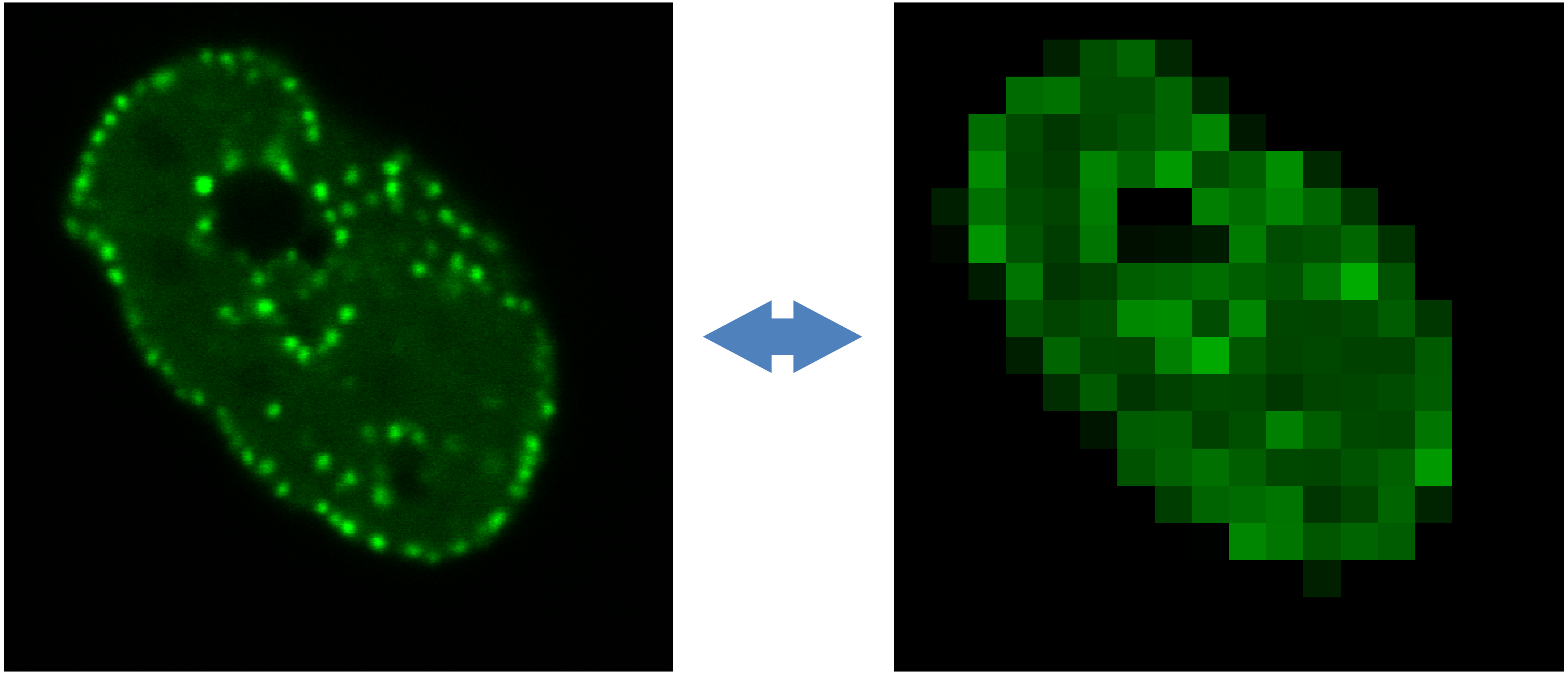
0	2	2	2	5	8	11	8	2	2	0	0	0	0	0	0	0	0
0	0	2	11	76	136	164	85	11	5	2	2	0	0	0	0	0	0
0	2	25	172	181	133	133	164	90	14	5	2	2	0	0	0	0	0
2	5	175	130	104	127	141	164	206	65	31	11	2	2	0	0	0	0
2	28	212	124	110	204	164	232	133	155	218	87	14	2	2	0	0	0
2	73	178	133	121	195	34	31	198	175	204	167	104	14	5	0	0	0
2	45	226	141	113	184	53	59	70	192	133	138	167	99	11	2	0	0
0	2	70	184	102	116	155	161	175	155	141	184	255	138	34	5	2	0
0	0	5	141	121	133	209	215	133	206	124	121	130	153	104	8	2	0
0	0	2	73	164	124	121	198	252	147	121	127	119	119	150	19	2	0
0	0	0	5	93	150	102	119	130	127	104	121	124	133	153	25	2	0
0	0	0	0	5	62	153	155	119	136	198	155	127	124	187	19	2	2
0	0	0	0	0	5	138	161	178	155	127	124	141	158	232	5	2	2
0	0	0	0	0	0	11	113	164	172	184	102	121	164	79	2	2	2
0	0	0	0	0	0	2	5	36	206	187	147	164	153	5	2	2	0
0	0	0	0	0	0	0	0	2	5	25	76	31	2	2	2	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Displaying a digital image

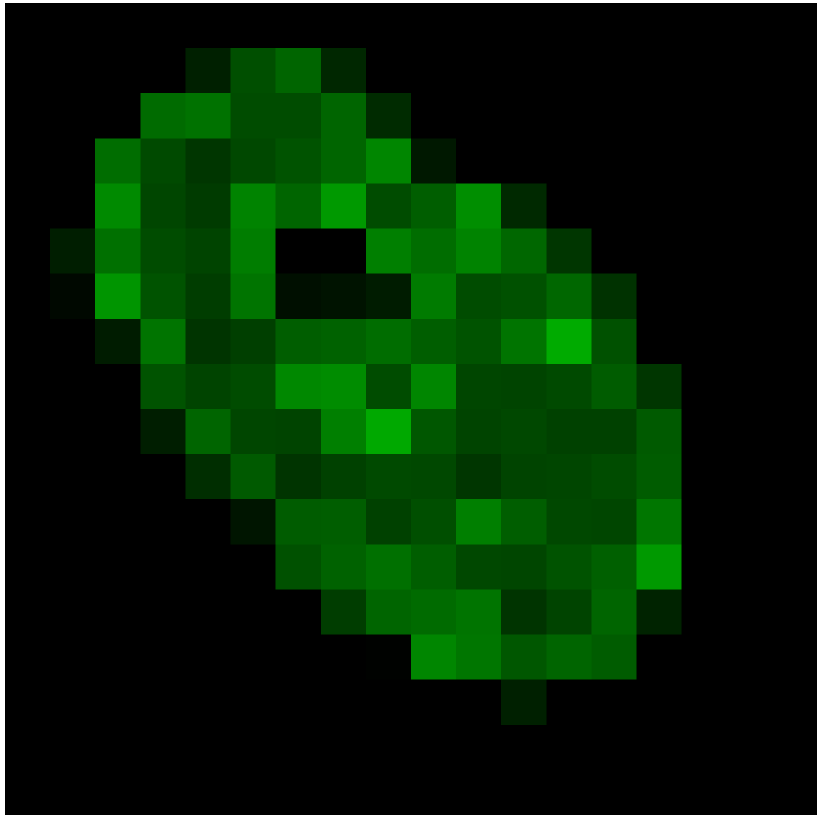
0	2	2	2	5	8	11	8	2	2	0	0	0	0	0	0	0	0
0	0	2	11	76	136	164	85	11	5	2	2	0	0	0	0	0	0
0	2	25	172	181	133	133	164	90	14	5	2	2	0	0	0	0	0
2	5	175	130	104	127	141	164	206	65	31	11	2	2	0	0	0	0
2	28	212	124	110	204	164	232	133	155	218	87	14	2	2	0	0	0
2	73	178	133	121	195	34	31	198	175	204	167	104	14	5	0	0	0
2	45	226	141	113	184	53	59	70	192	133	138	167	99	11	2	0	0
0	2	70	184	102	116	155	161	175	155	141	184	255	138	34	5	2	0
0	0	5	141	121	133	209	215	133	206	124	121	130	153	104	8	2	0
0	0	2	73	164	124	121	198	252	147	121	127	119	119	150	19	2	0
0	0	0	5	93	150	102	119	130	127	104	121	124	133	153	25	2	0
0	0	0	0	5	62	153	155	119	136	198	155	127	124	187	19	2	2
0	0	0	0	0	5	138	161	178	155	127	124	141	158	232	5	2	2
0	0	0	0	0	0	11	113	164	172	184	102	121	164	79	2	2	2
0	0	0	0	0	0	2	5	36	206	187	147	164	153	5	2	2	0
0	0	0	0	0	0	0	0	2	5	25	76	31	2	2	2	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



Comparing the original and digital image



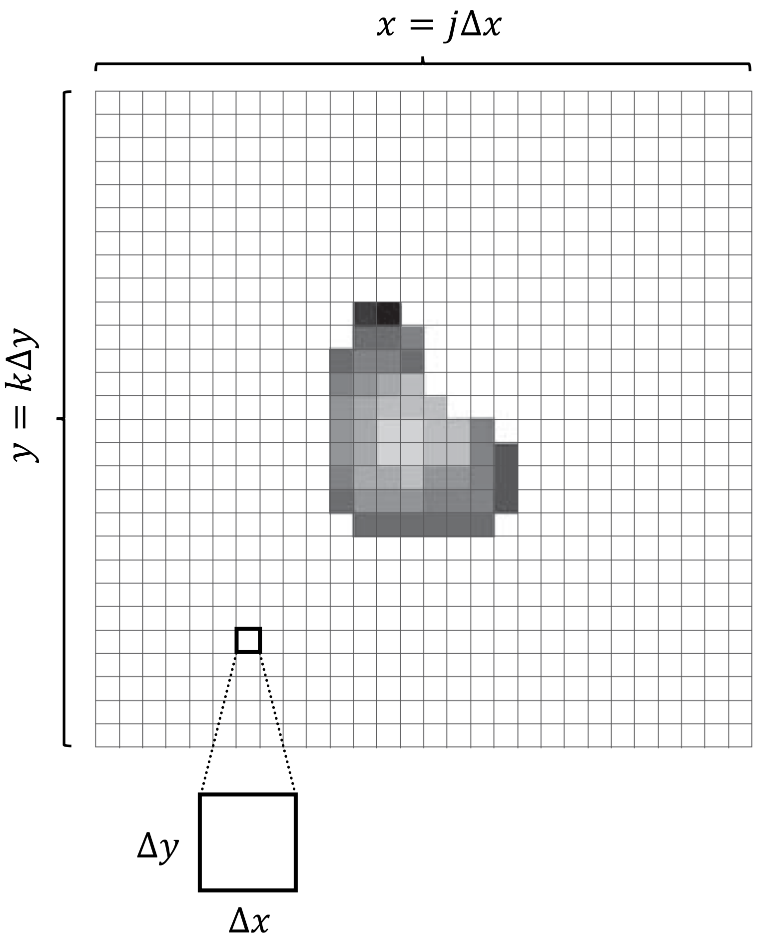
Comparing the original and digital image



Digitisation by spatial sampling

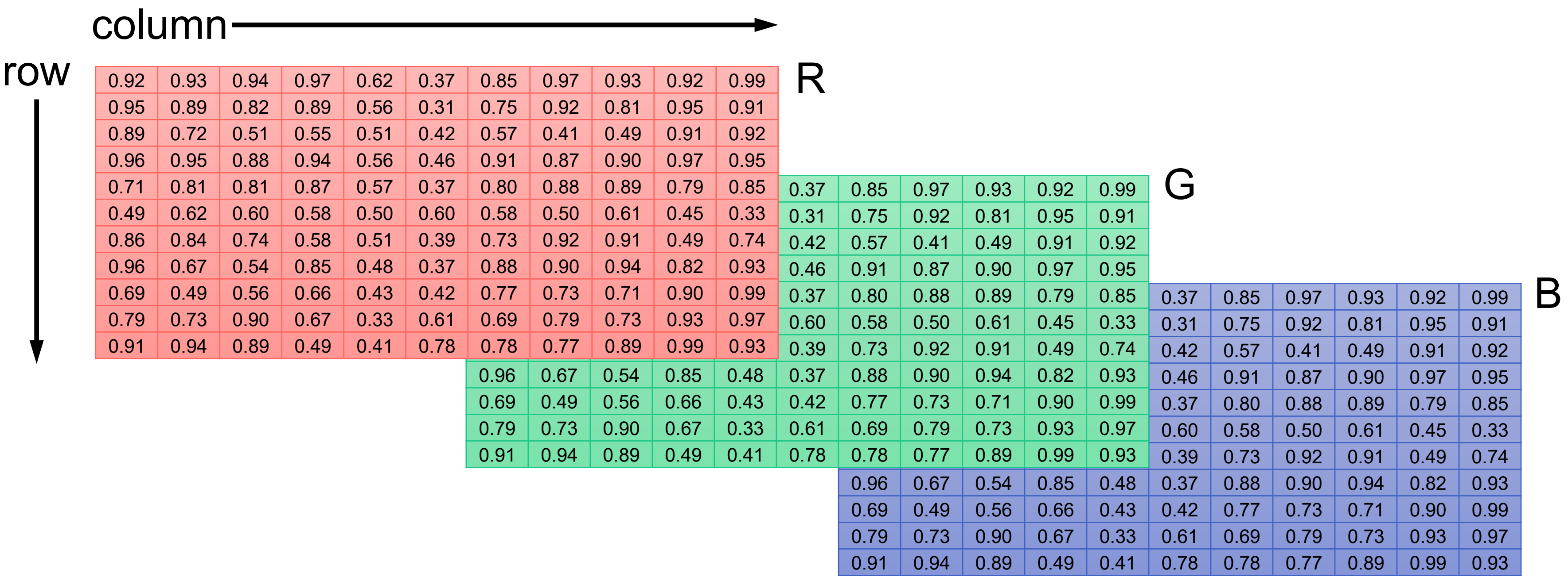
- Digitisation converts an analog image to a digital image by sampling the image space
- Sampling discretises the coordinates x and y
  - Spatial discretisation of a picture function $f(x, y)$
  - Typically a rectangular grid of sampling points is used

    $x = j\Delta x,\ y = k\Delta y$ for $j = 0 \ldots M-1,\ k = 0 \ldots N-1$
  - The Δx and Δy are called the sampling intervals



Digital colour images

Each channel is a digital image with the same number of rows and columns



Spatial resolution

- Spatial resolution: number of pixels per unit of length
- Example: resolution decreases by one half each time (see right)
- Human faces can be recognized at 64 x 64 pixels per face
- Appropriate resolution is essential:
  - Too little resolution yields poor recognition
  - Too much resolution is slow and wastes memory

1/1

1/2

1/4

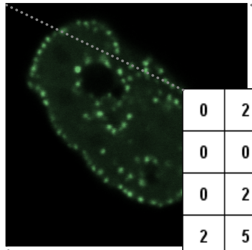
1/8

1/16

Quantisation

- Quantisation digitises the image intensity or amplitude values $f(x, y)$
  - Called intensity or gray-level quantisation
  - Gray-level resolution to be chosen per application
    - ❖ For example, 16, 32, 64, ..., 128, 256 levels
    - ❖ Should be high enough for human perception of shading details
    - ❖ The latter requires about 100 levels for a realistic image
    - ❖ Should not be higher than necessary to avoid wasting storage

Quantisation and bits per pixel



0	2	2	2	5	8	11	8	2	2	0	0	0	0	0	0	0	0
0	0	2	11	76	136	164	85	11	5	2	2	0	0	0	0	0	0
0	2	25	172	181	133	133	164	90	14	5	2	2	0	0	0	0	0
2	5	175	130	104	127	141	164	206	65	31	11	2	2	0	0	0	0
2	28	212	124	110	204	164	232	133	155	218	87	14	2	2	0	0	0
2	73	178	133	121	195	34	31	198	175	204	167	104	14	5	0	0	0
2	45	226	141	113	184	53	59	70	192	133	138	167	99	11	2	0	0
0	2	70	184	102	116	155	161	175	155	141	184	255	138	34	5	2	0
0	0	5	141	121	133	209	215	133	206	124	121	130	153	104	8	2	0
0	0	2	73	164	124	121	198	252	147	121	127	119	119	150	19	2	0
0	0	0	5	93	150	102	119	130	127	104	121	124	133	153	25	2	0
0	0	0	0	5	62	153	155	119	136	198	155	127	124	187	19	2	2
0	0	0	0	0	5	138	161	178	155	127	124	141	158	232	5	2	2
0	0	0	0	0	0	11	113	164	172	184	102	121	164	79	2	2	2
0	0	0	0	0	0	2	5	36	206	187	147	164	153	5	2	2	0
0	0	0	0	0	0	0	0	2	5	25	76	31	2	2	2	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Pixel (picture element)

Levels per pixel:

8 bits = 2^8 = 256

12 bits = 2^{12} = 4,096

16 bits = 2^{16} = 65,536

24 bits = 2^{24} = 16,777,216

Further reading on discussed topics

- Chapter 2 of Szeliski
- Chapter 2 of Shapiro and Stockman

Acknowledgements

- Several slides from Derek Hoiem, Alexei Efros, Steve Seitz, and David Forsyth
- Some material drawn from referenced and associated online sources
- Image sources credited where possible