GRAPHICS GROUP

IVS Intelligent Vision Systems New Zealand

# Computer Graphics and Image Processing

## Part 3: Image Processing

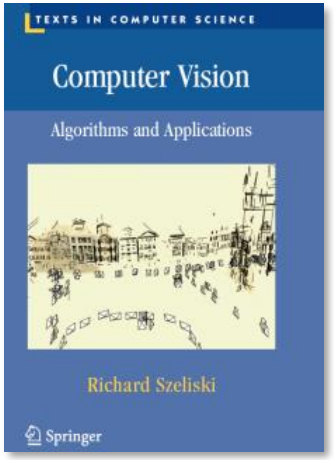
## 1 – Digital Images & Histograms

*Martin Urschler, PhD*

# Credits

- Based on previous year's lecture slides prepared by Prof. Georgy Gimel'farb and Assoc Prof Patrice Delmas

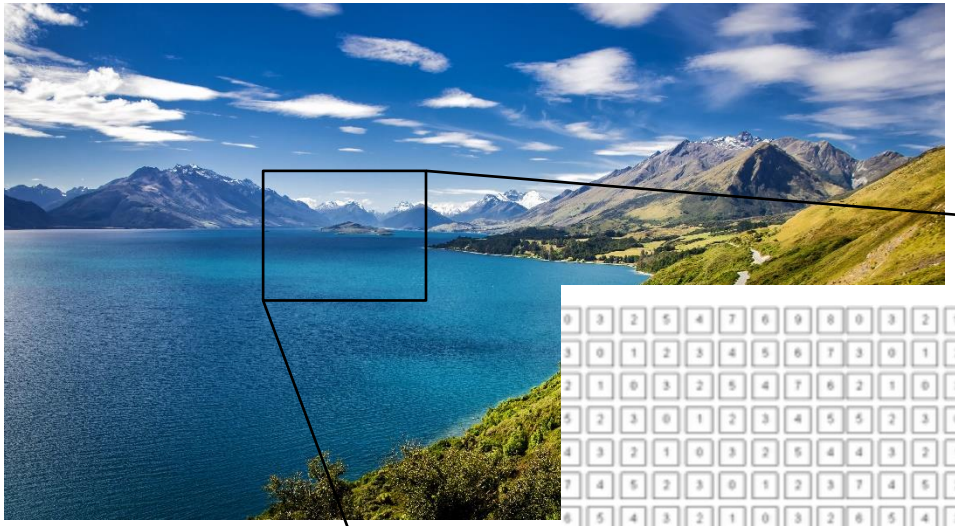
- Selected Material from:
  - Lecture slides by Richard Alan Peters, Noah Snavely, Steve Seitz
  - Book "Computer Vision: Algorithms and Applications" by Rick Szeliski
  - Book "Digital Image Processing" by R. Gonzalez and R. Woods



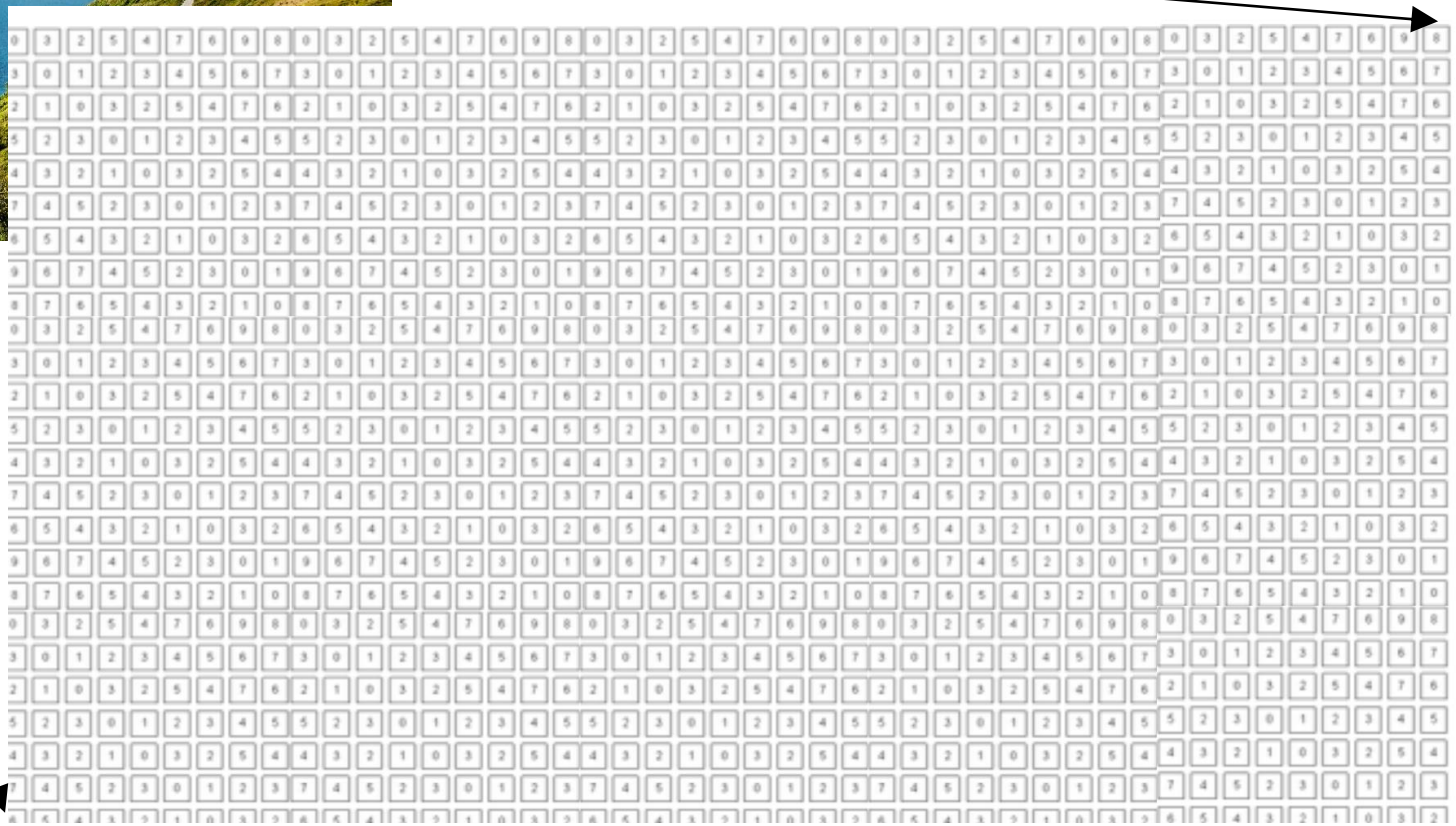

online at: http://szeliski.org/Book/

# What is a digital image?

# What is a digital image?

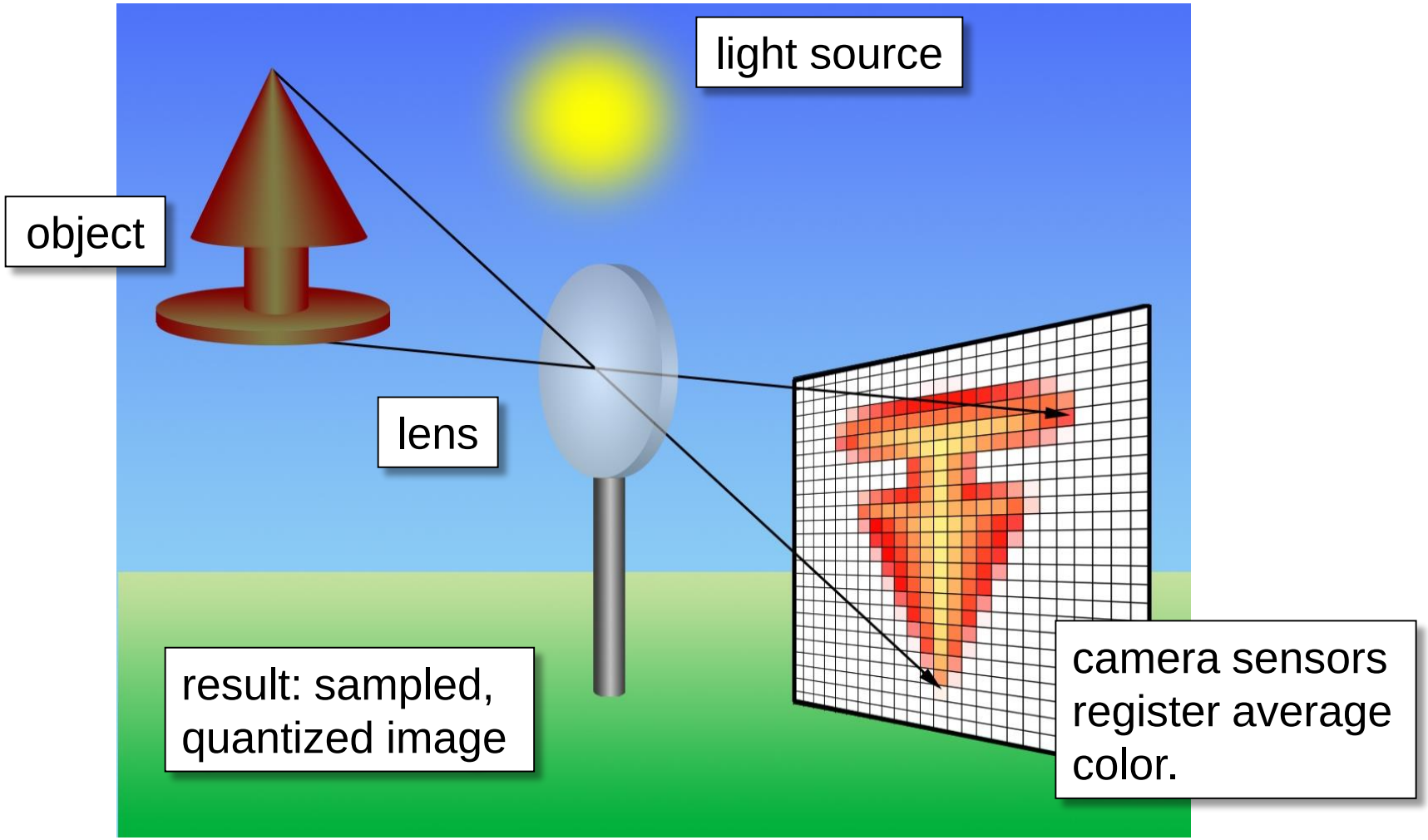


A matrix of numbers



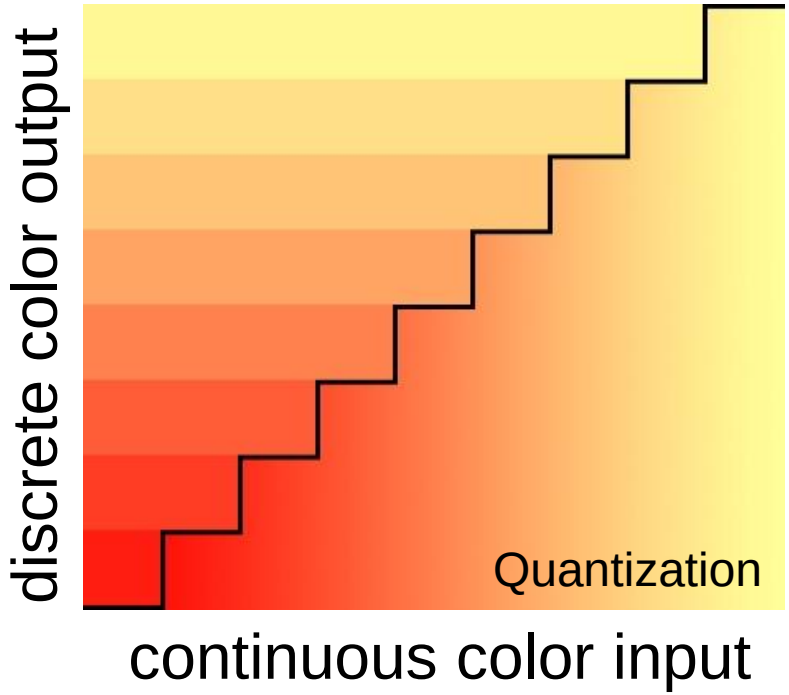
# Image formation

- (Very) simplified camera model

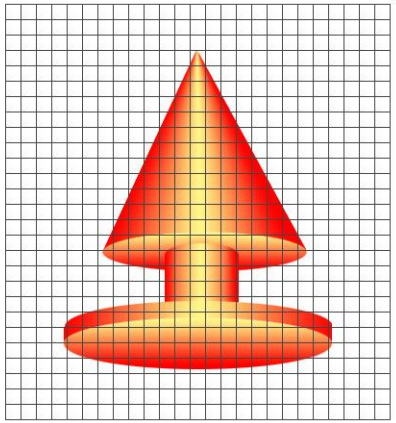


# Image formation

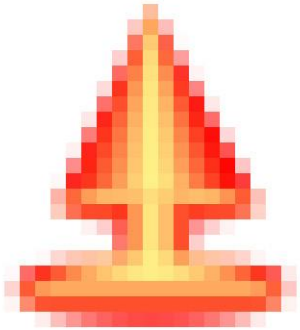
- Spatial sampling
- Quantization
  - Continuous colors mapped to finite, discrete set of colors




pixel grid



real image



sampled

quantized

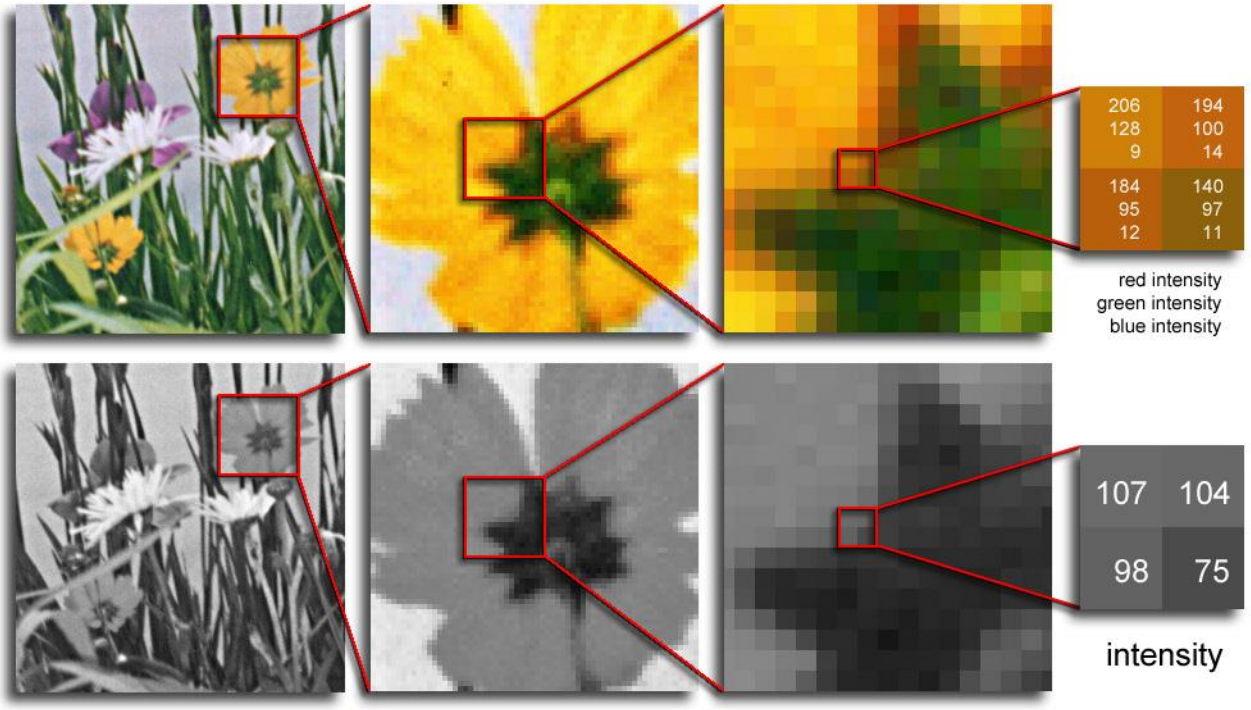
sampled & quantized

# Digital image

Color images have 3 values per pixel; monochrome images have 1 value per pixel.

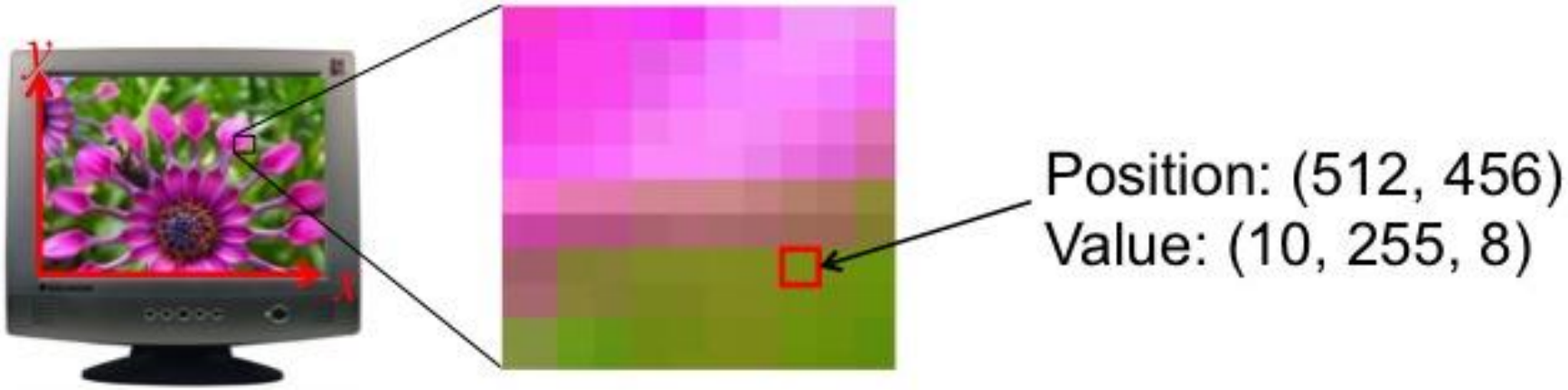
a grid of squares, each of which contains a single color

each square is called a pixel (for *picture element*)



# Digital image

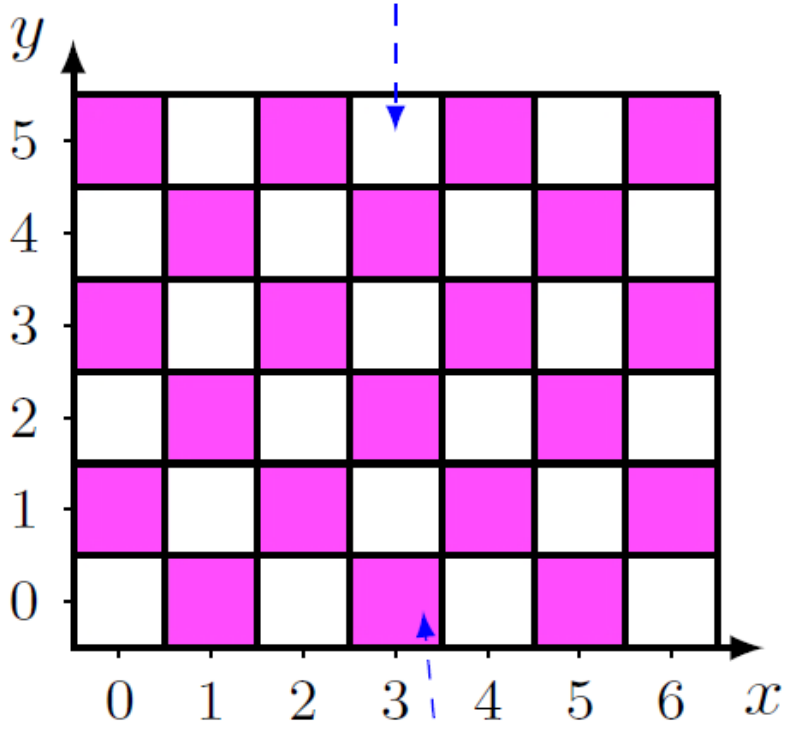
- Pixels, image resolution, coordinate systems




  - Origin (0,0) of pixel coordinates is sometimes in top left corner
  - Image resolution: how many pixels? [width x height]
  - Warning on origin and axes:
    - Origin not always consistent
    - Some software (Matlab) may not accept 0 as a valid position index

# Colour digital image 6x7 pixels

Colour "white": $v = [\underset{R}{255}, \underset{G}{255}, \underset{B}{255}]$

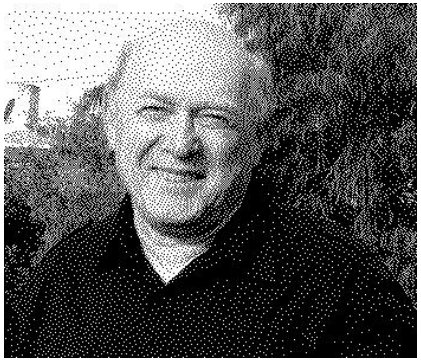



Colour "magenta": $v = [\underset{R}{255}, \underset{G}{0}, \underset{B}{255}]$

2D vector array $\mathbf{V} = [v(x, y) : x = 0, \ldots, 6; y = 0, \ldots, 5]$:

| $y$ | $x$=0 | 1 | 2 | ... |
|---|---|---|---|---|
| 0 | $v(0,0)$ | $v(1,0)$ | $v(2,0)$ | ... |
| 1 | $v(0,1)$ | $v(1,1)$ | $v(2,1)$ | ... |
| ... | ... | ... | ... | ... |
| 5 | $v(0,5)$ | $v(1,5)$ | $v(2,5)$ | ... |

# Encoding of colours



Prof. Georgy Gimel'farb

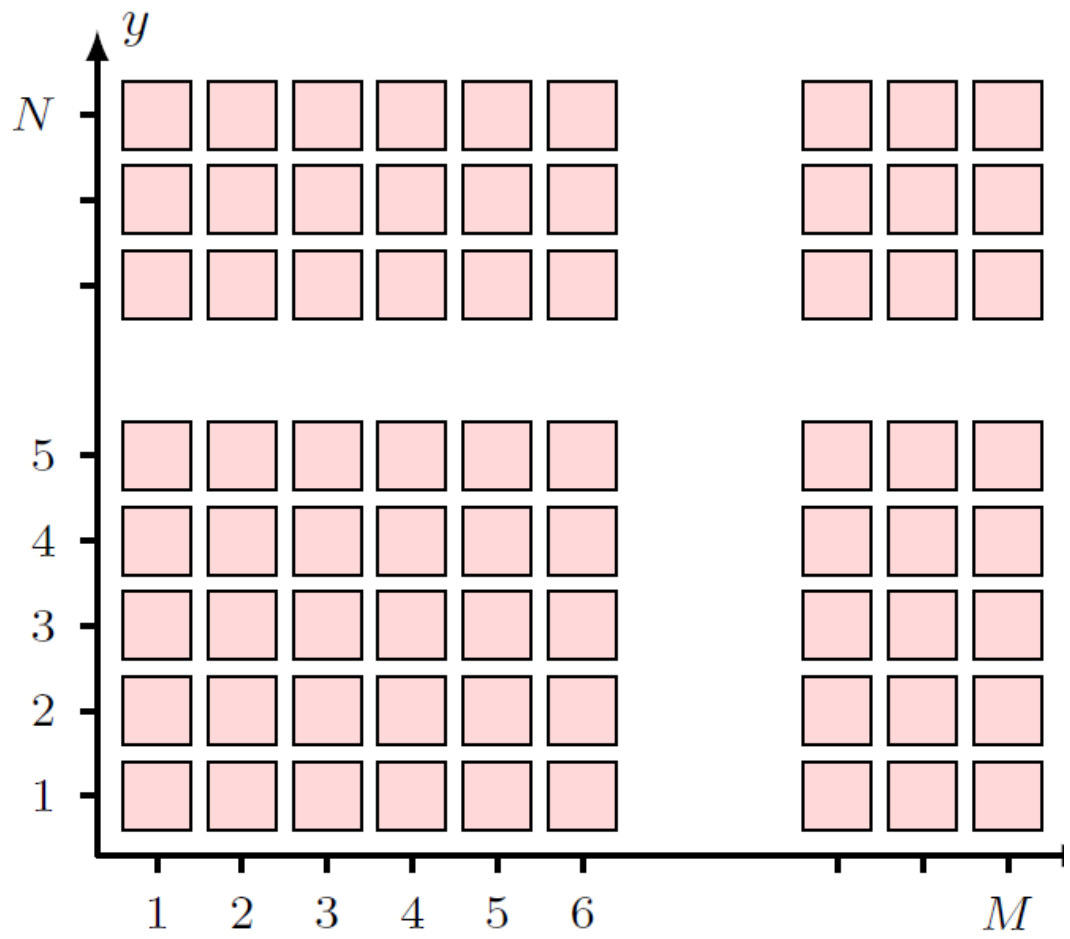
- **Bit depth**: a number of bits used to represent each pixel's value (typically 1,8,24,or 32)
  - Binary image (1 bit / pixel)
    - Only two codes – 0 (black) and 1 (white)
  - Scalar/monochrome/greyscale image (8 bits)
    - Scalar codes: just a single number per colour (index into palette)
    - No colour – grey values from black to white
  - Vector-valued image: vector codes (24 bits or more)
    - Several (e.g. three) numbers per pixel: R,G,B
    - Huge number of colours can be represented
- This course considers mostly binary and greyscale images!

# Defining images mathematically

An image can be defined on an $M \times N$ arithmetic grid, or lattice:

$$\mathbf{R}_{M,N} = \{(x, y) : \ x = 1, 2, \ldots, M; \ y = 1, 2, \ldots, N\}$$

- $(x, y)$ – pixel coordinates.
- An image as a graph of a function $f : \mathbf{R} \to \mathbf{V}$.
- $\mathbf{V}$ – a set of signal values, e.g. grey levels or colours.
- **Example**: pixel at position $(100, 50)$ has scalar value 255, i.e. $f(100, 50) = 255$.



# Python Excursion

- Reading and displaying an image with Python 3 and matplotlib

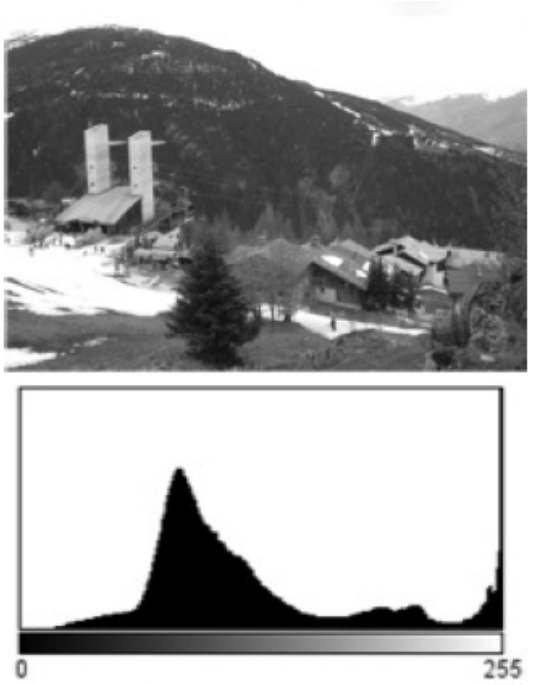
# Image histogram

**The distribution (or empirical probabilities, or counts) of the pixel intensities in an image.**

The histogram gives numbers of pixels at each different intensity value found in that image.

- For an 8-bit image, each pixel has an intensity value between 0 and 255.
- For a 16-bit image, each pixel has an intensity value between 0 and 65,535.
- For a colour image with 8-bit RGB channels, each pixel has one of 16,777,216 different colour values.

$2^8 = 256$; $2^{16} = 65,536$; $2^{24} = 16{,}777{,}216$




| | |
|---|---|
| Count: | 1080000 |
| Min: | 10 |
| Max: | 255 |
| Mean: | 146.9 |
| StdDev: | 73.6 |
| Mode: | 255 |
| Mode count: | 236885 |

# Image histogram

- An $M \times N$ greyscale 8-bit image $v$ ($M$ rows and $N$ columns) has $K = M \times N$ pixels.
- Each pixel $(x, y)$ has an integer intensity $v(x, y)$ in the range $\mathbf{Q} = [0, 1, \ldots, 255]$.

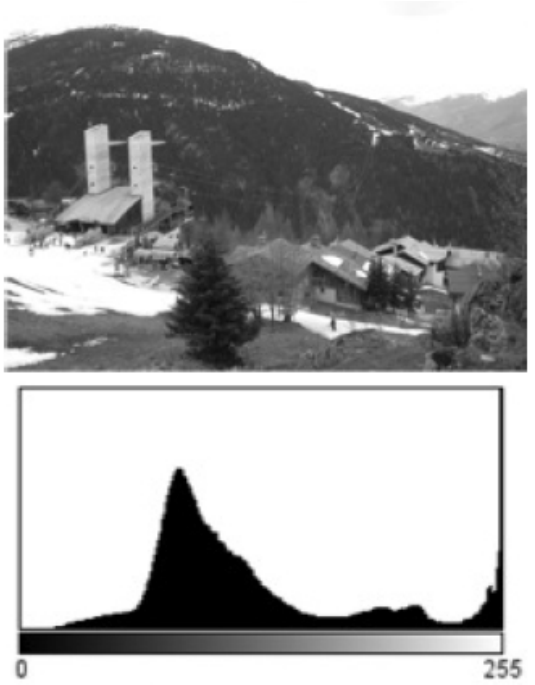
The histogram $H : \mathbf{Q} \to [0, 1, \ldots, K]$ for the image $v$ records the numbers, $H(q)$, of pixels with intensities $v(x, y) = q \in \mathbf{Q}$:

$$H(q) = \sum_{(x,y)=(0,0)}^{(M,N)} \delta(v(x, y) - q);\ q \in \mathbf{Q}$$

$$H(0) + H(1) + \ldots + H(255) = K$$

---

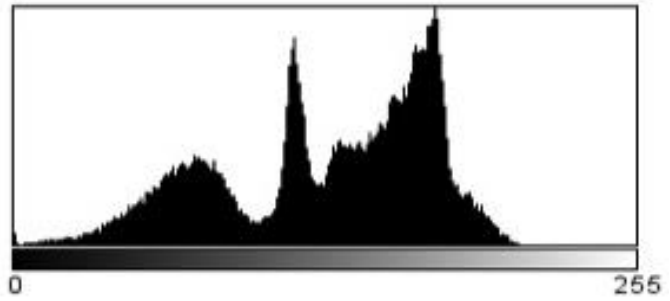
Kronecker's $\delta$-function: $\delta(z) = 1$ for $z = 0$ and $= 0$ otherwise.

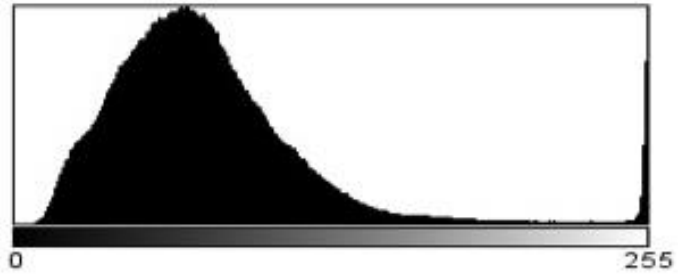



| | |
|---|---|
| Count: | 1080000 |
| Min: | 10 |
| Max: | 255 |
| Mean: | 146.9 |
| StdDev: | 73.6 |
| Mode: | 255 |
| Mode count: | 236885 |

# Computing the histogram

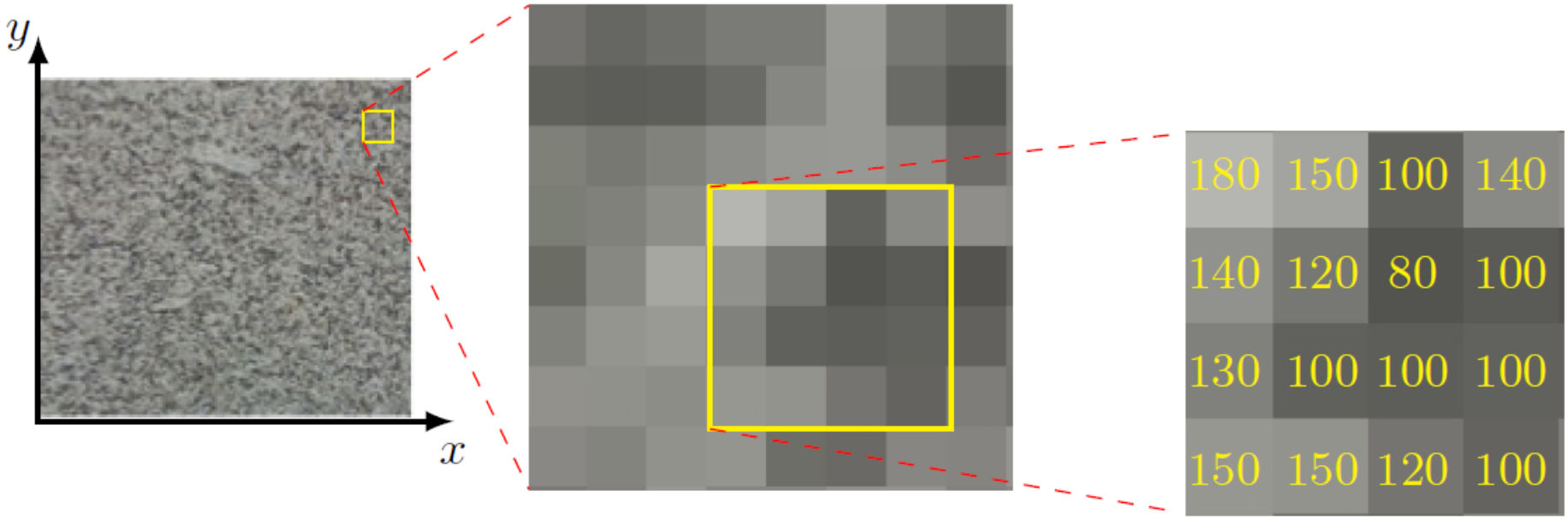
1. The image is scanned in a single pass
2. A running count of the number of pixels at each intensity is kept
3. These values are graphed to visualize the histogram



# Image histogram – An example

Let's look at a magnified portion of an image:

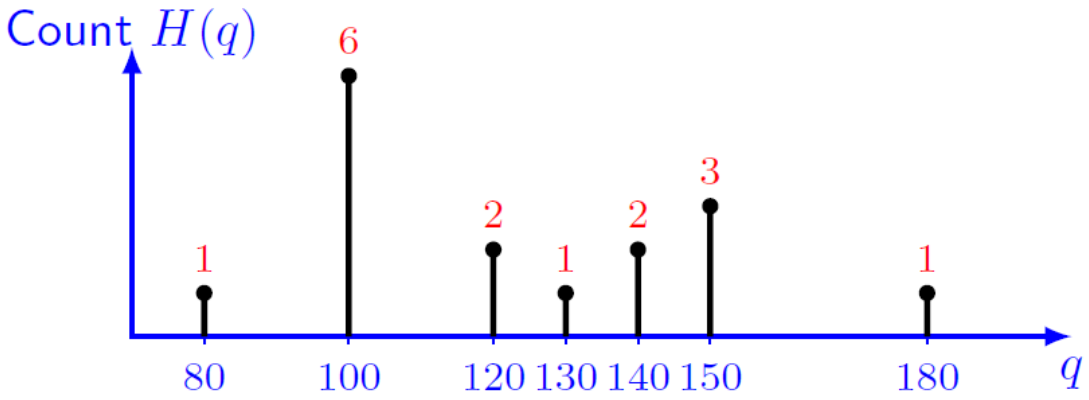



Update of the histogram $H$ after scanning the right $4 \times 4$ portion:

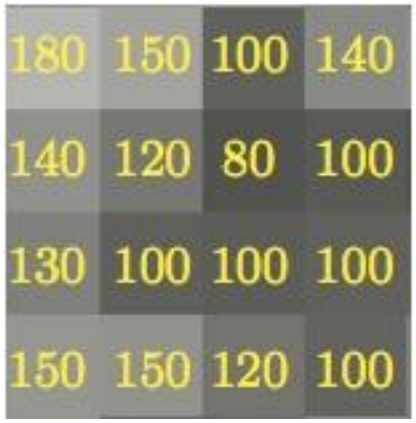
$$H(80) \leftarrow H(80) + 1; \quad H(100) \leftarrow H(100) + 6; \quad H(120) \leftarrow H(120) + 2;$$
$$H(130) \leftarrow H(130) + 1; \quad H(140) \leftarrow H(140) + 2; \quad H(150) \leftarrow H(150) + 3;$$
$$H(180) \leftarrow H(180) + 1$$

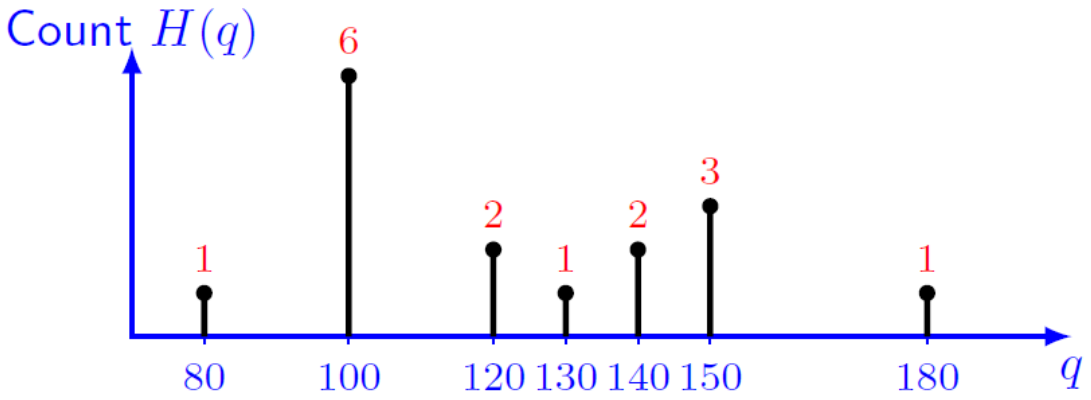
# Image histogram – An example

| | | | |
|---|---|---|---|
| 180 | 150 | 100 | 140 |
| 140 | 120 | 80 | 100 |
| 130 | 100 | 100 | 100 |
| 150 | 150 | 120 | 100 |

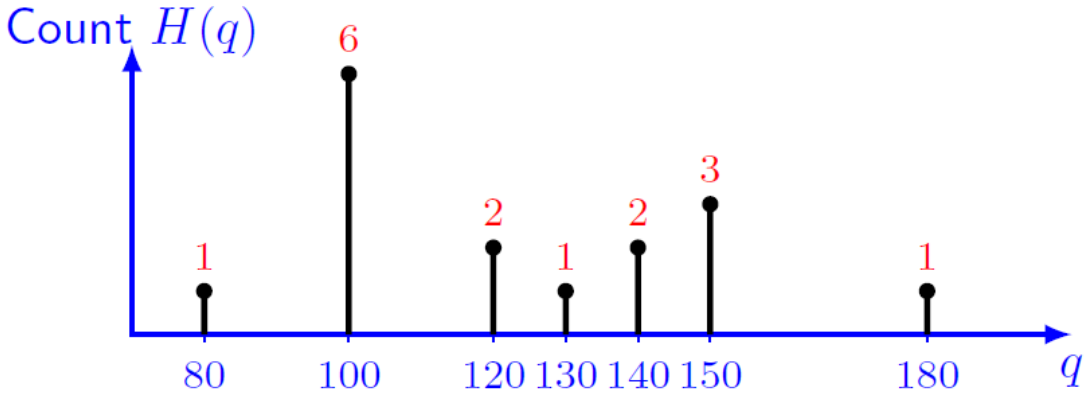


# Image histogram – An example



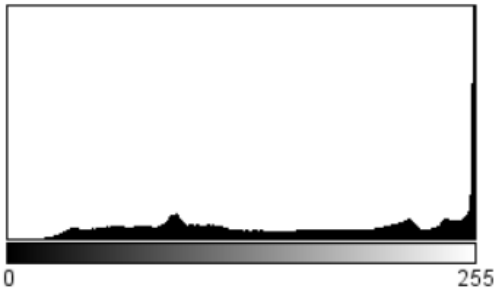




Is an image defined uniquely by its histogram?

# Histograms of Under-/Over-Exposed Photos

Underexposed

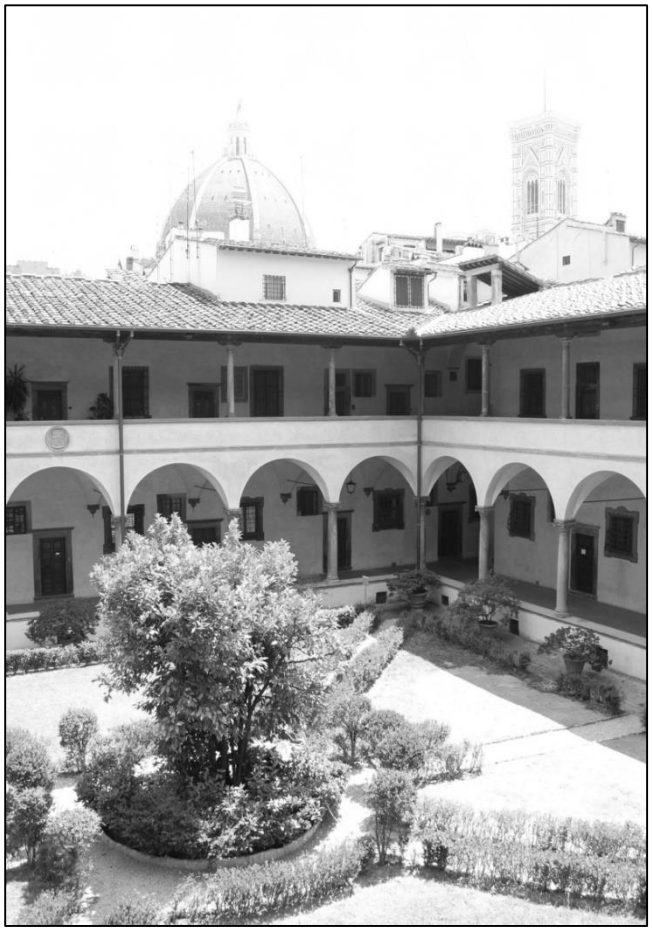



Count: 918400
Mean: 172.751
StdDev: 76.924

Min: 10
Max: 255
Mode: 255 (200682)


Count: 918400
Mean: 63.235
StdDev: 66.709

Min: 0
Max: 251
Mode: 2 (40205)



Overexposed

https://www.easyhdr.com/examples

# Python Excursion

- Computing and displaying a histogram of an image