

---

# VC 12/13 – T4

## Colour and Noise

Mestrado em Ciência de Computadores  
Mestrado Integrado em Engenharia de Redes e  
Sistemas Informáticos

***Miguel Tavares Coimbra***

---

# Outline

---

- Colour spaces
- Colour processing
- Noise

---

# Topic: Colour spaces

---

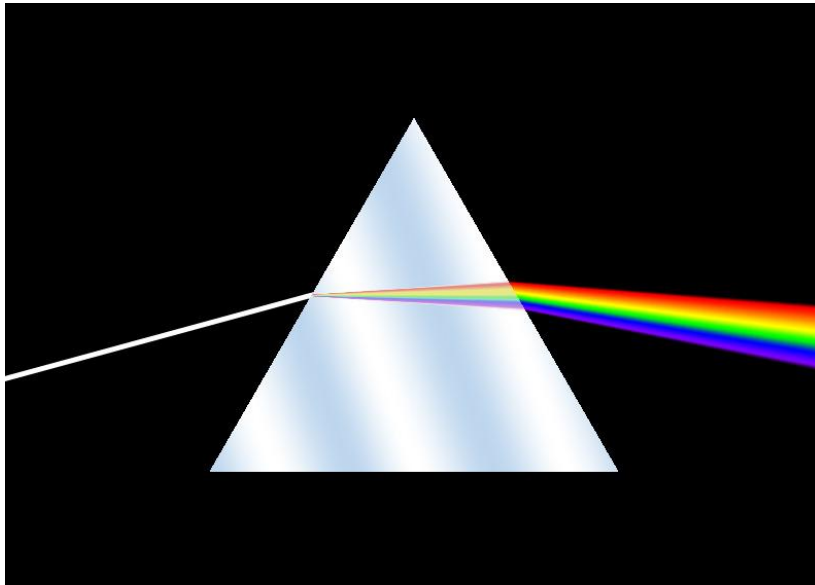
- Colour spaces
- Colour processing
- Noise



For a long  
time I limited  
myself to one  
colour – as a  
form of  
discipline

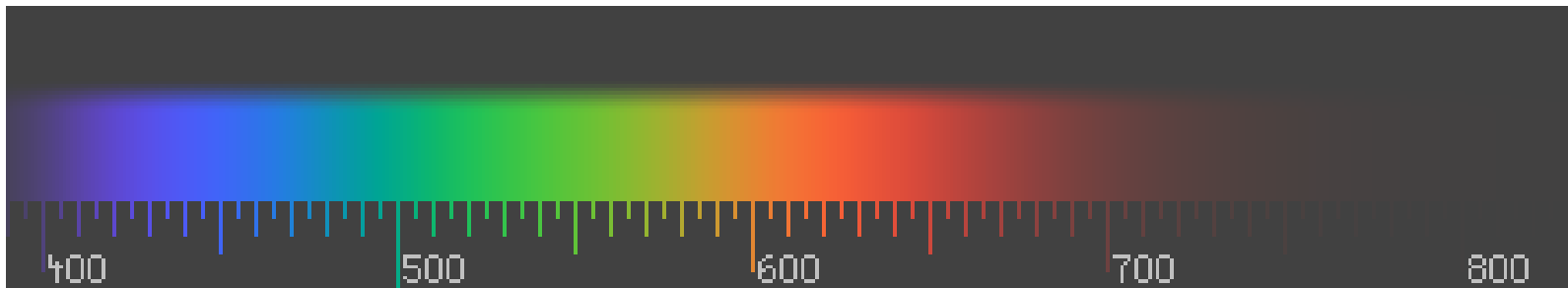
*Pablo  
Picasso*

# What is colour?

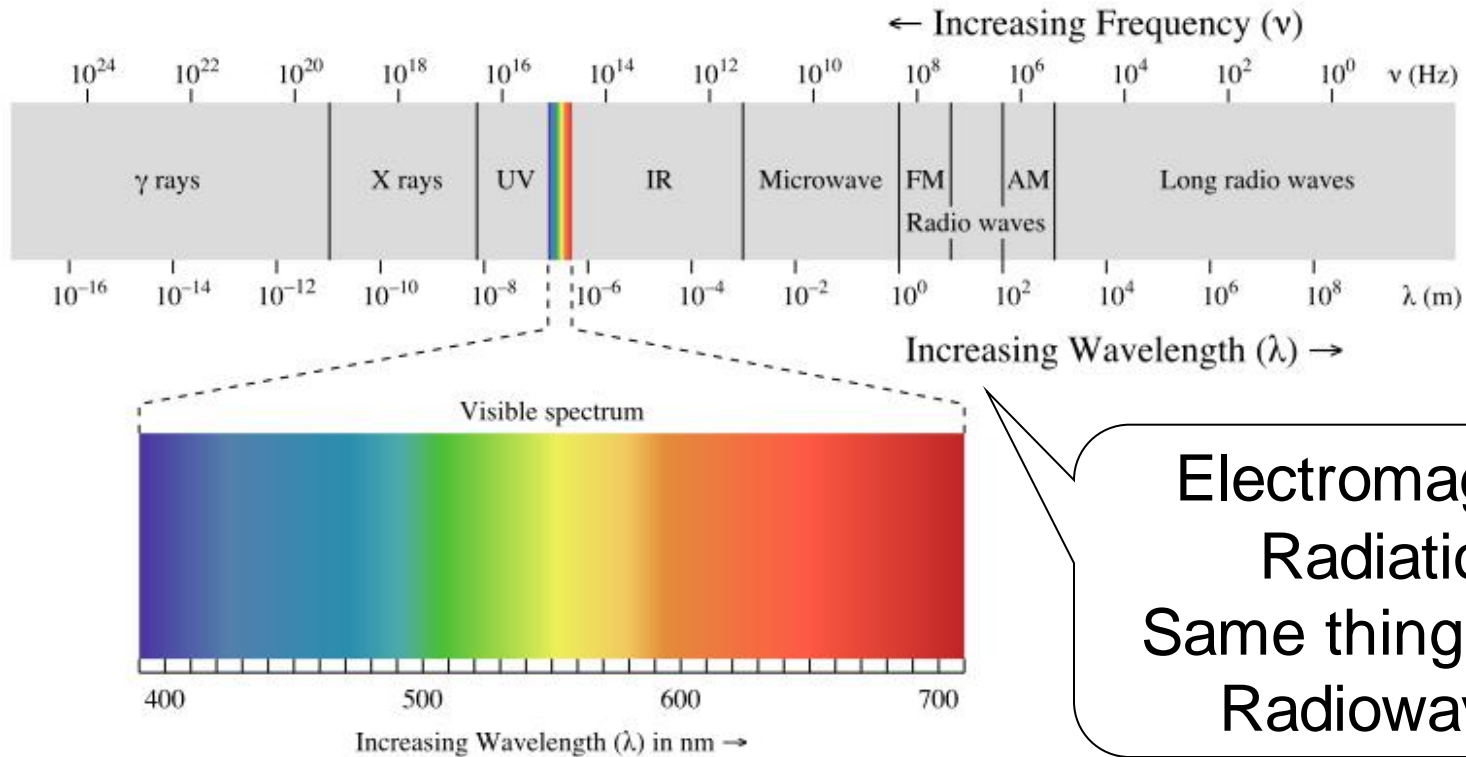


Optical Prism  
dispersing light

Visible colour  
spectrum

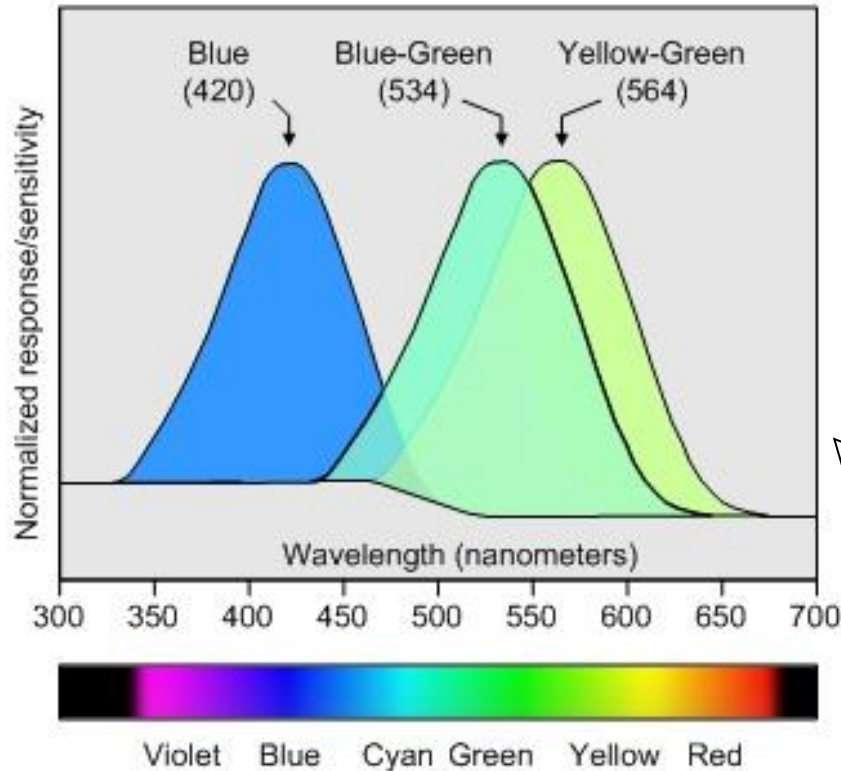


# Visible Spectrum



<http://science.howstuffworks.com/light.htm>

# How do we see colour?



Human  
Colour  
Sensors:  
Cones

65% 'Red' cones  
33% 'Green' cones  
2% 'Blue' cones

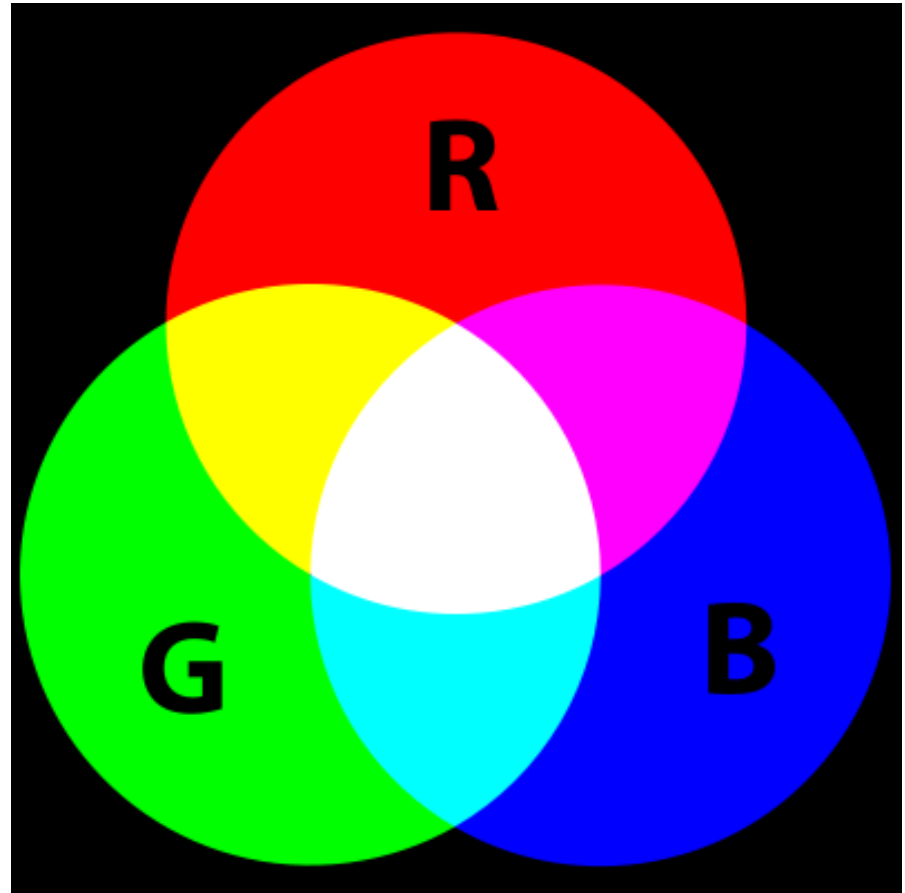
Typical humans are trichromats  
(three color cone/pigment types – blue, blue-green, and yellow-green)

---

# Primary Colours

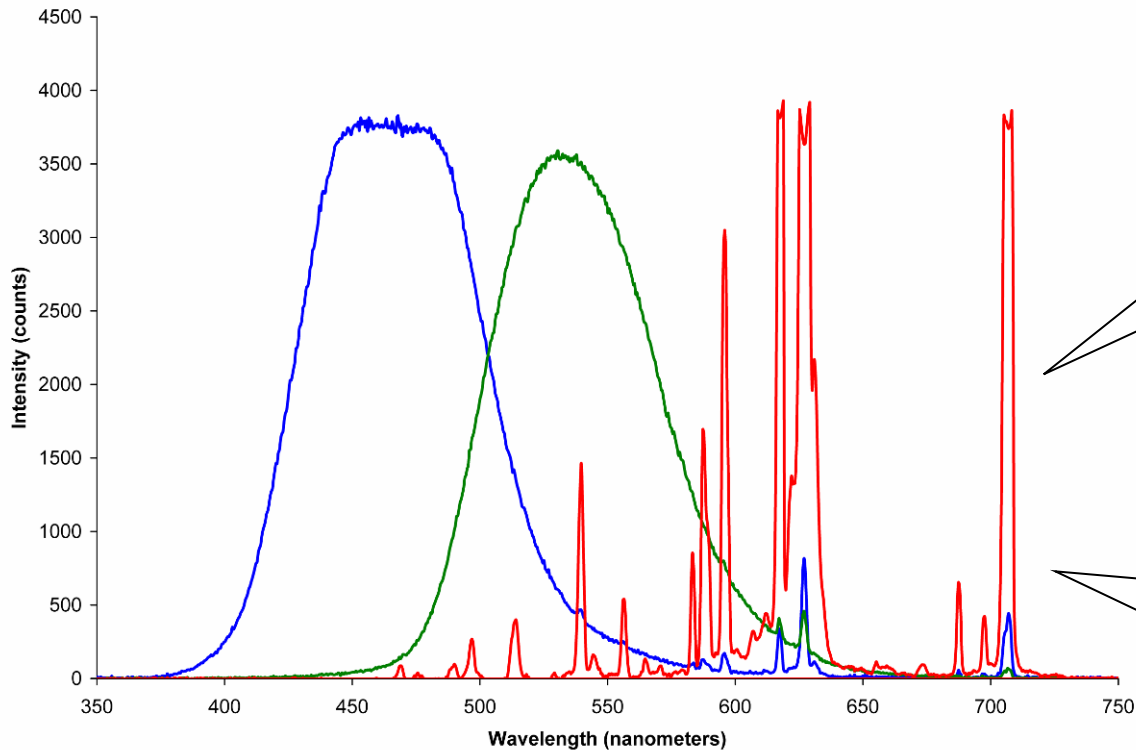
---

- Not a fundamental property of light.
- Based on the physiological response of the human eye.
- Form an additive colour system.





# Example: Television



Three types of phosphors very close together

The components are added to create a final colour

<http://www.howstuffworks.com/tv.htm>

---

# Colour Space

---

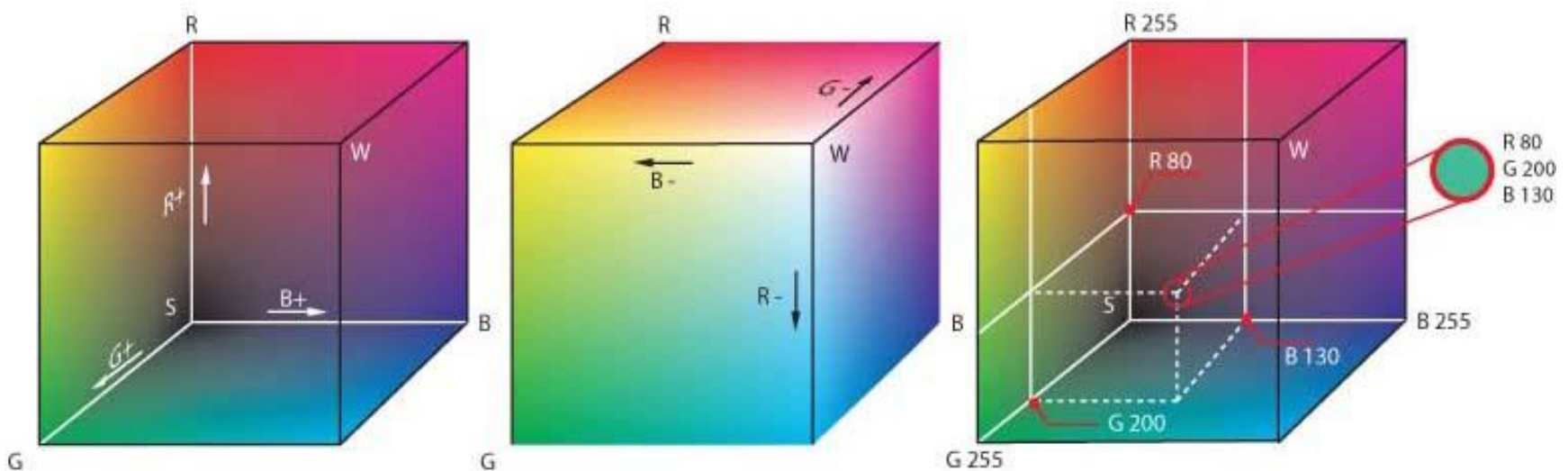
- “The purpose of a color model is to facilitate the specification of colours in some standard, generally accepted way”

Gonzalez & Woods

- **Colour space**
  - Coordinate system
  - Subspace: One colour -> One point

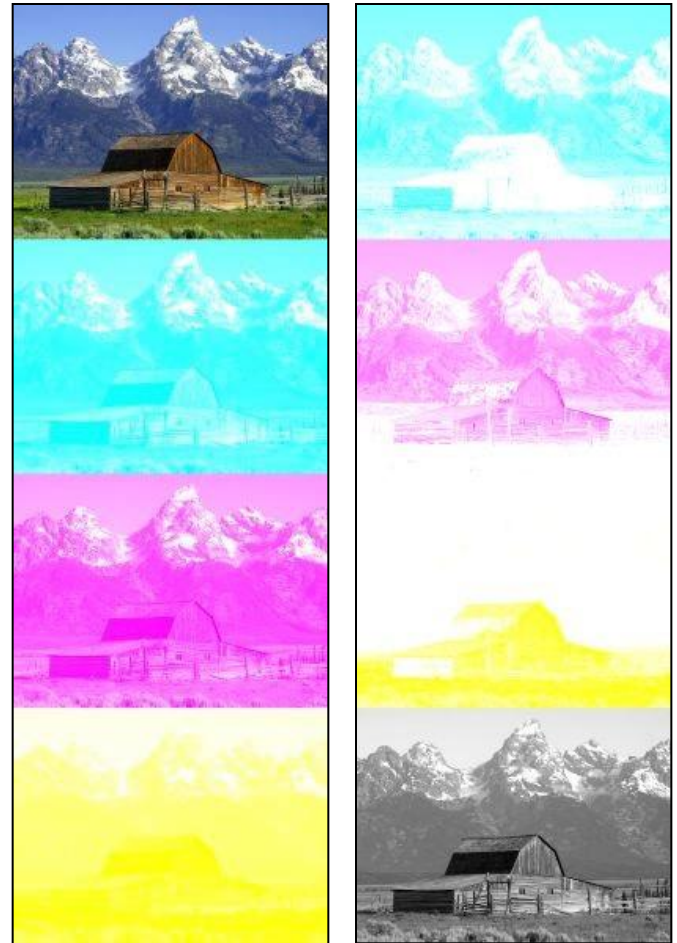
# RGB

- Red Green Blue
- Defines a colour cube.
- Additive components.
- Great for image capture.
- Great for image projection.
- Poor colour description.



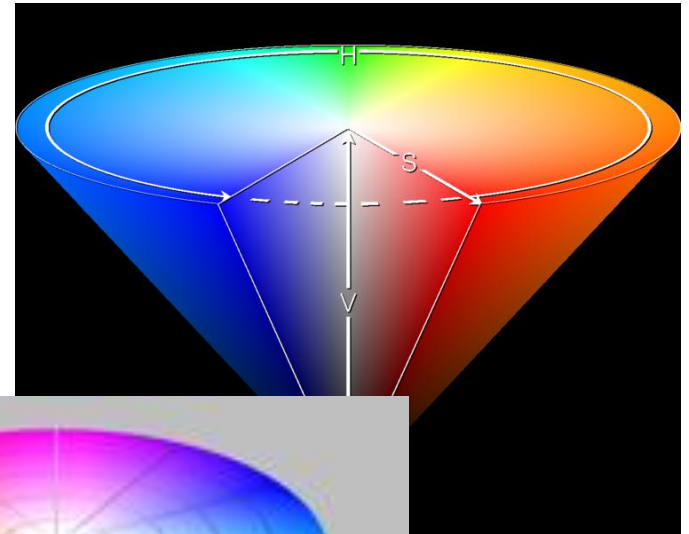
# CMYK

- Cyan Magenta Yellow Key.
- Variation of RGB.
- Technological reasons: great for printers.



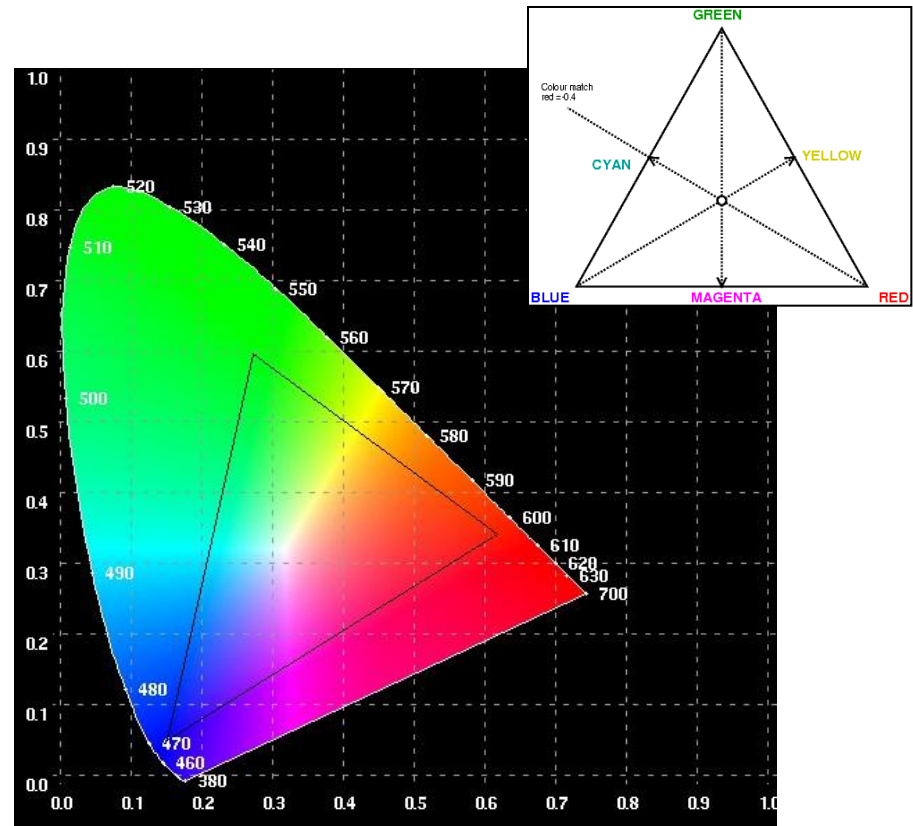
# HSI

- Hue Saturation  
Intensity
- Defines a colour cone
- Great for colour description.



# Chromaticity Diagram

- **Axis:**
  - Hue
  - Saturation
- Outer line represents our visible spectrum.
- No three primaries can create all colours!

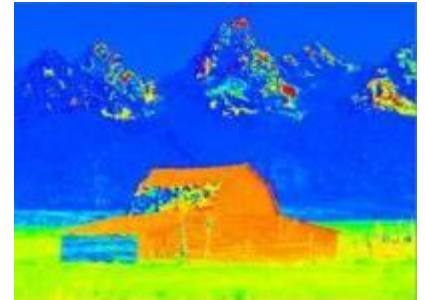


[http://www.cs.rit.edu/~ncs/color/a\\_chroma.html](http://www.cs.rit.edu/~ncs/color/a_chroma.html)

# RGB to HSI

Hue:

$$H = \begin{cases} \theta & \Leftarrow B \leq G \\ 360 - \theta & \Leftarrow B > G \end{cases}$$



$$\theta = \cos^{-1} \left\{ \frac{\frac{1}{2} [(R - G) + (R - B)]}{\left[ (R - G)^2 + (R - B)(G - B) \right]^{1/2}} \right\}$$



Saturation

$$S = 1 - \frac{3}{(R + G + B)} [\min(R, G, B)]$$



Intensity

$$I = \frac{1}{3} (R + G + B)$$

# HSI to RGB

- Depends on the 'sector' of H

$$120 \leq H < 240$$

$$0 \leq H < 120$$

$$B = I(1 - S)$$

$$R = I \left[ 1 + \frac{S \cos H}{\cos(60^\circ - H)} \right]$$

$$G = 3I - (R + B)$$

$$240 \leq H < 360$$

$$H = H - 120^\circ$$

$$R = I(1 - S)$$

$$G = I \left[ 1 + \frac{S \cos H}{\cos(60^\circ - H)} \right]$$

$$B = 3I - (R + G)$$

$$H = H - 240^\circ$$

$$G = I(1 - S)$$

$$B = I \left[ 1 + \frac{S \cos H}{\cos(60^\circ - H)} \right]$$

$$R = 3I - (G + B)$$



---

# Topic: Colour processing

---

- Colour spaces
- **Colour processing**
- Noise



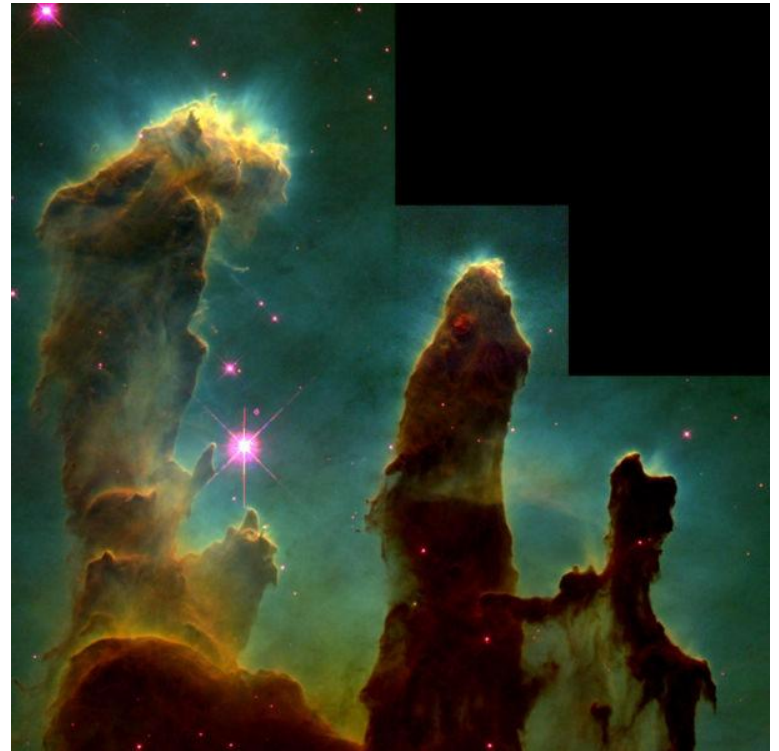
A WFPC2 image of a small region of the [Tarantula Nebula](#) in the [Large Magellanic Cloud](#) [NASA/ESA]

---

# Pseudocolour

---

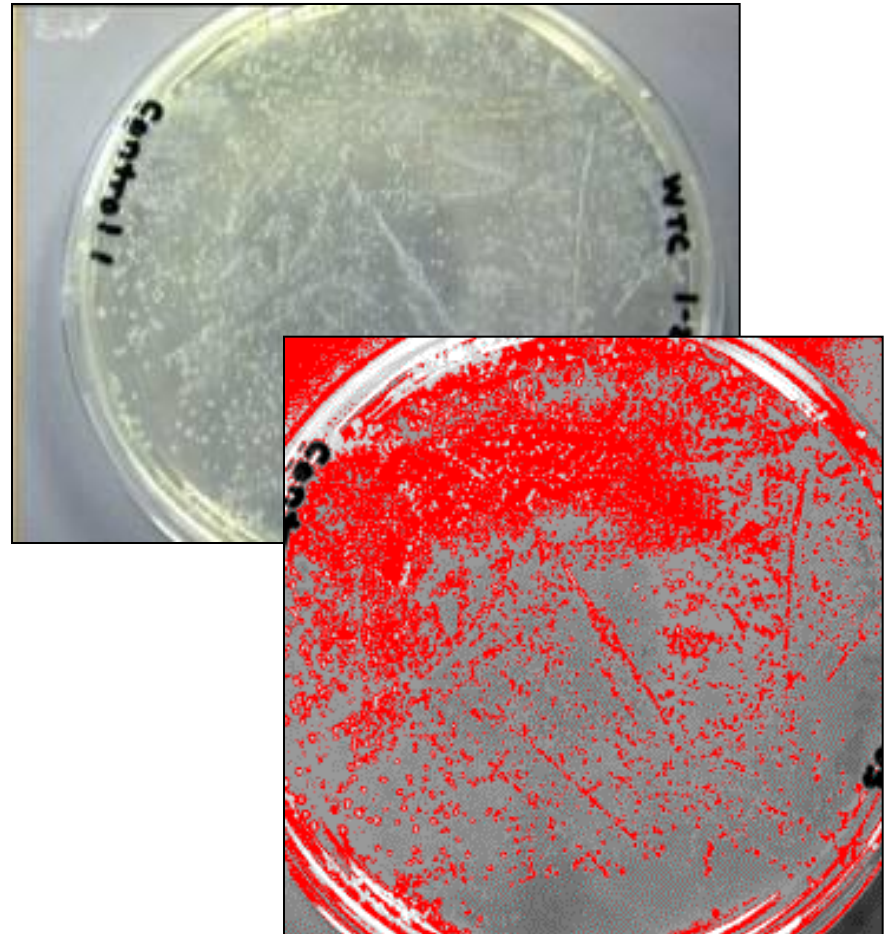
- Also called *False Colour*.
- Opposed to *True Colour* images.
- The colours of a pseudocolour image do not attempt to approximate the real colours of the subject.

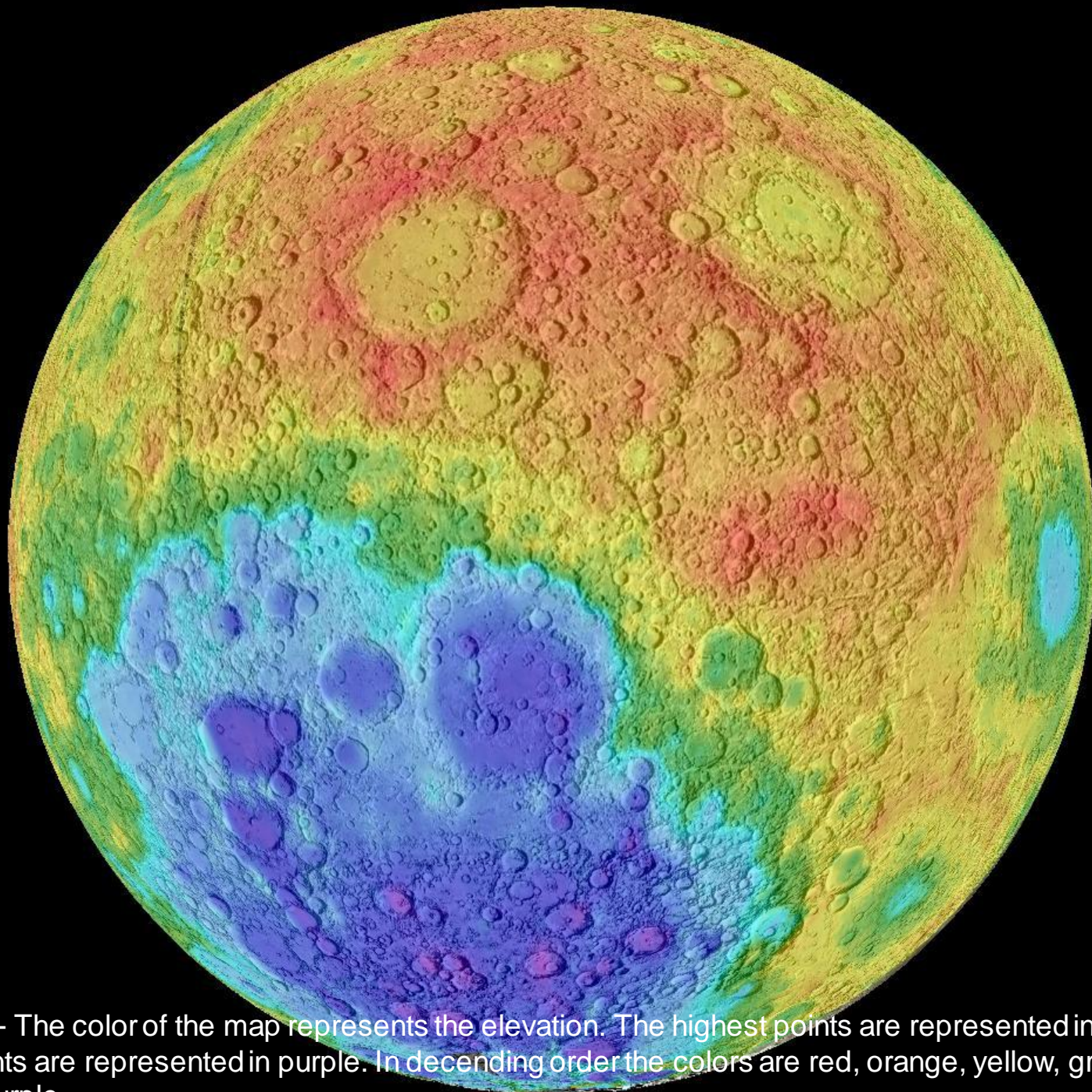


One of Hubble's most famous images: "pillars of creation" where stars are forming in the Eagle Nebula. [NASA/ESA]

# Intensity Slicing

- Quantize pixel intensity to a specific number of values (*slices*).
- Map one colour to each *slice*.
- Loss of information.
- Enhanced human visibility.

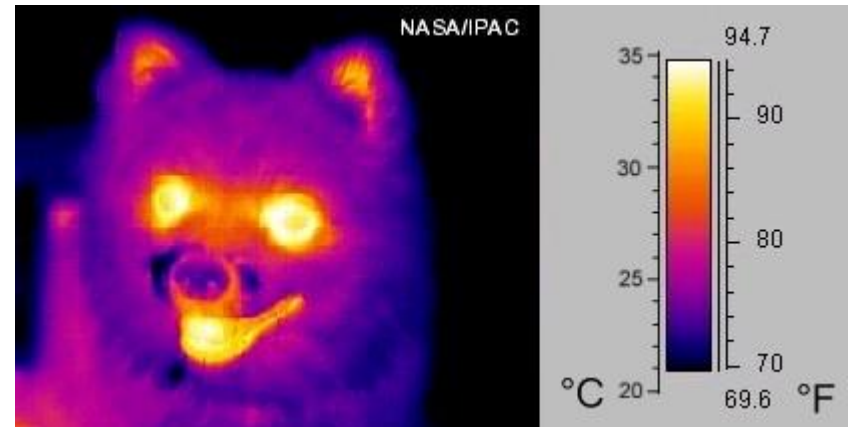




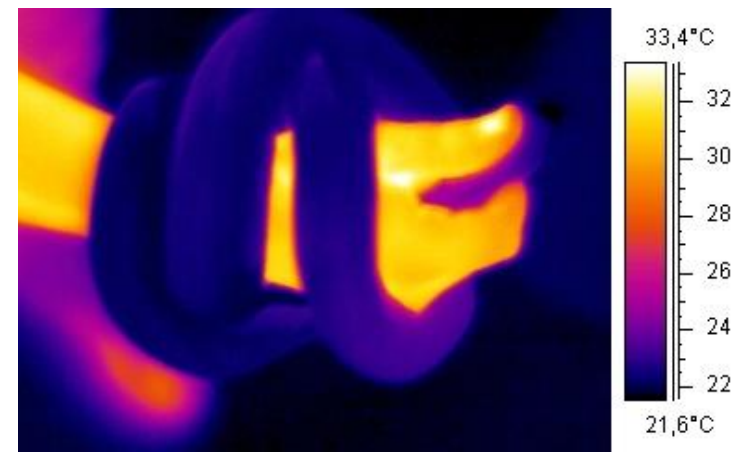
The [Moon](#) - The color of the map represents the elevation. The highest points are represented in red. The lowest points are represented in purple. In descending order the colors are red, orange, yellow, green, cyan, blue and purple.

# Intensity to Colour Transformation

- Each colour component is calculated using a transformation function.
- Viewed as an Intensity to Colour map.
- Does not need to use RGB space!



$$f(x, y) \Rightarrow \begin{cases} f_R(x, y) = T_R[f(x, y)] \\ f_G(x, y) = T_G[f(x, y)] \\ f_B(x, y) = T_B[f(x, y)] \end{cases}$$



A supernova  
remnant created  
from the death of  
a massive star  
about 2,000 years  
ago.



[http://chandra.harvard.edu/photo/false\\_color.html](http://chandra.harvard.edu/photo/false_color.html)

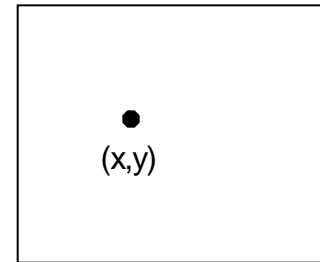
<http://landsat.gsfc.nasa.gov/education/compositor/>

# Colour Image Processing

- **Grey-scale image**

- One value per position.

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

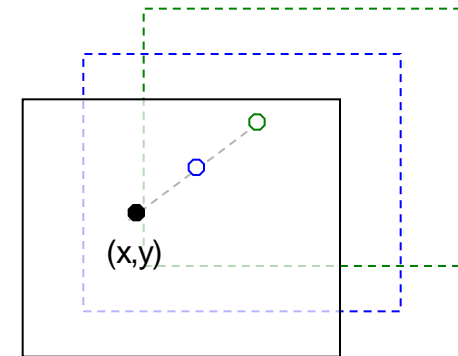


Grey-scale image

- **Colour image**

- One vector per position.

$$f(x,y) = [R \ G \ B]^T$$



RGB Colour image



# Colour Transformations

- Consider single-point operations:

$T_i$ : Transformation function for colour component  $i$

$s_i, r_i$ : Components of  $g$  and  $f$

$$g(x, y) = T[f(x, y)]$$

$$s_i = T_i(r_1, r_2, \dots, r_n)$$

$$i = 1, 2, \dots, n$$

- Simple example:

- Increase Brightness of an RGB image

$$s_R = r_R + 20$$

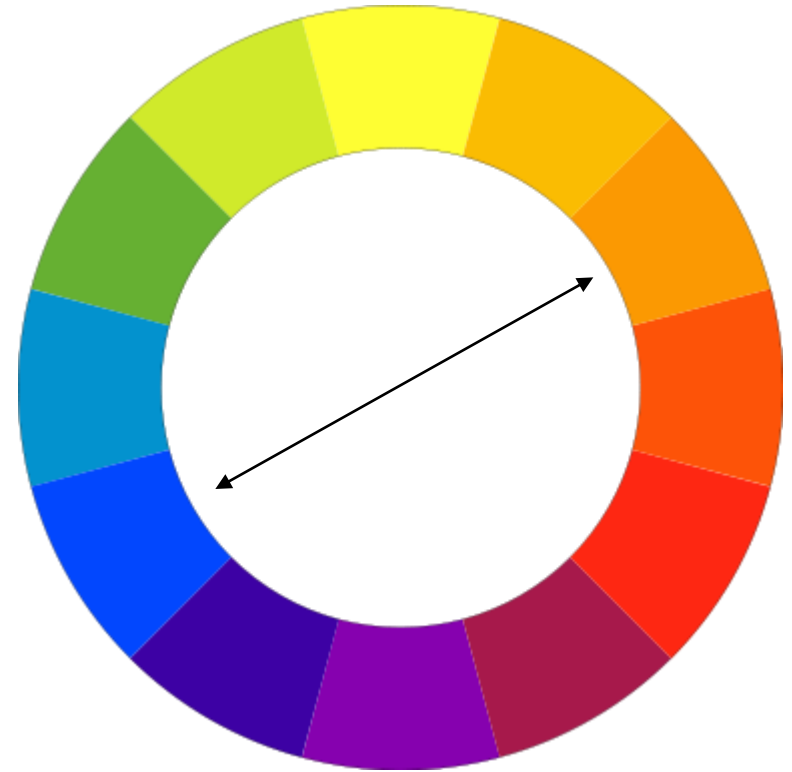
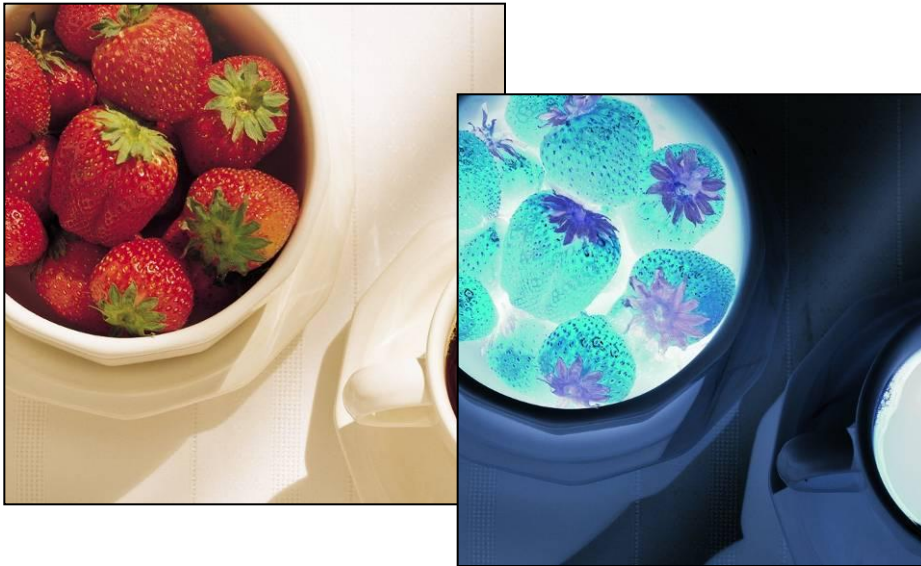
$$s_G = r_G + 20$$

$$s_B = r_B + 20$$

What about an image negative?

# Colour Complements

- Colour equivalent of an image negative.



Complementary Colours

---

# Colour Slicing

---

- Define a hyper-volume of interest inside my colour space.
- Keep colours if inside the hyper-volume.
- Change the others to a neutral colour.



---

# Topic: Noise

---

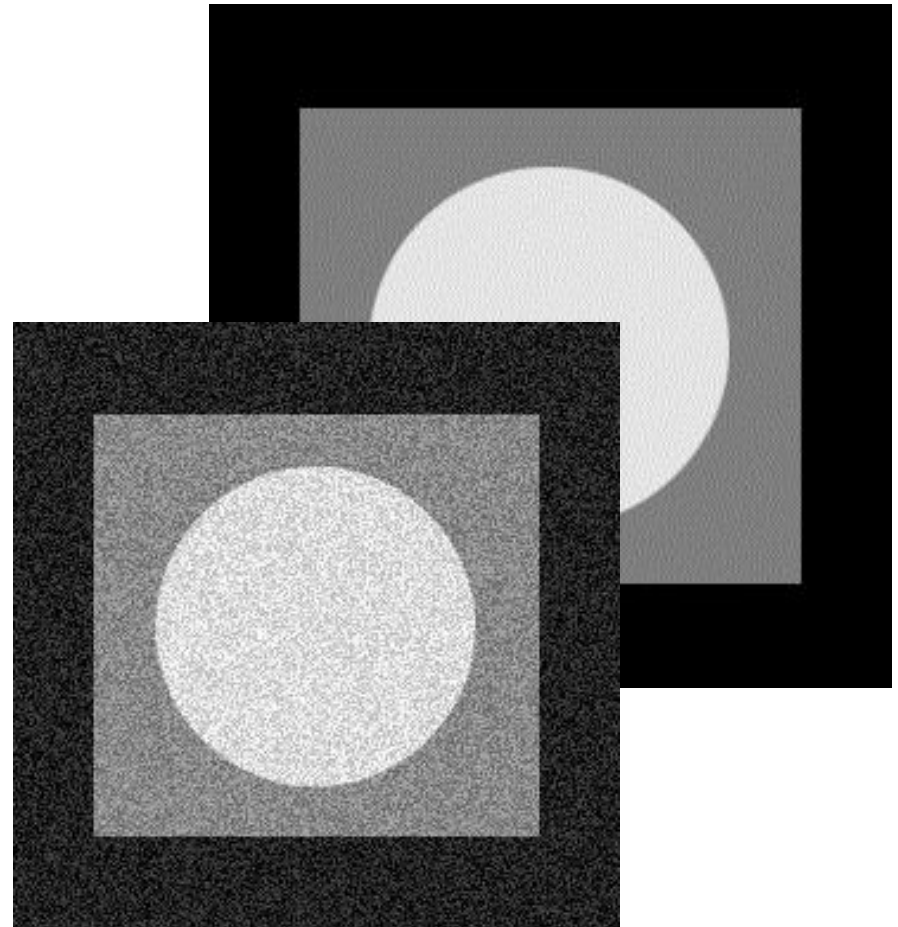
- Colour spaces
- Colour processing
- **Noise**

---

# Bring the Noise

---

- Noise is a distortion of the measured signal.
- Every physical system has noise.
- Images:
  - The importance of noise is affected by our human visual perception
  - Ex: Digital TV ‘block effect’ due to noise.



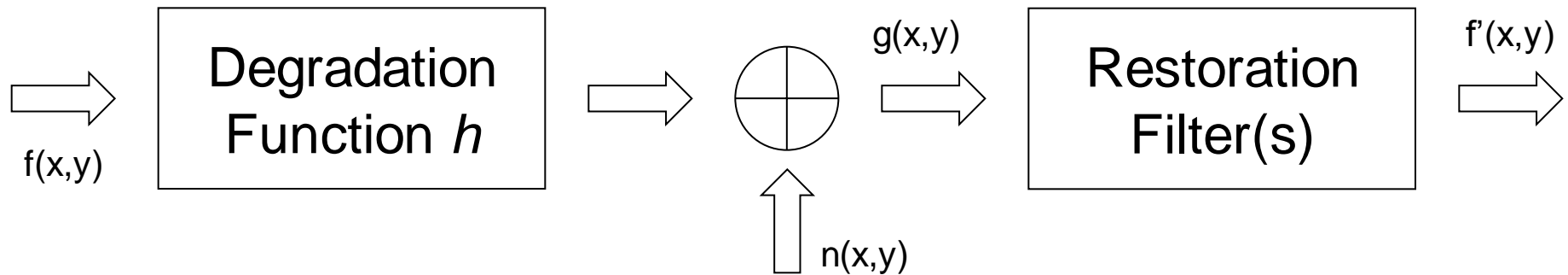
---

# Where does it come from?

---

- ‘Universal’ noise sources:
  - Thermal, sampling, quantization, measurement.
- Specific for digital images:
  - The number of photons hitting each images sensor is governed by quantum physics:  
*Photon Noise.*
  - Noise generated by electronic components of image sensors:
    - *On-Chip Noise, KTC Noise, Amplifier Noise, etc.*

# Degradation / Restoration



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

$$G(u, v) = H(u, v)F(u, v) + N(u, v)$$

---

# Noise Models

---

- **Noise models**
  - We need to mathematically handle noise.
  - Spatial and frequency properties.
  - Probability theory helps!
- **Advantages:**
  - Easier to filter noise.
  - Easier to measure its importance.
  - More robust systems!

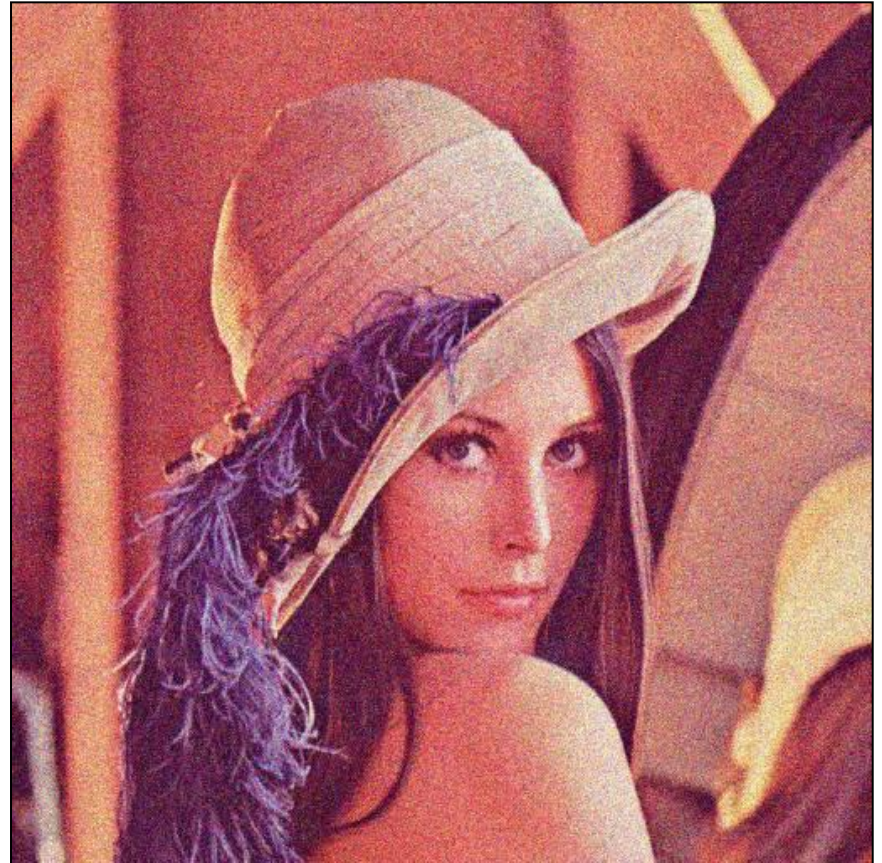


---

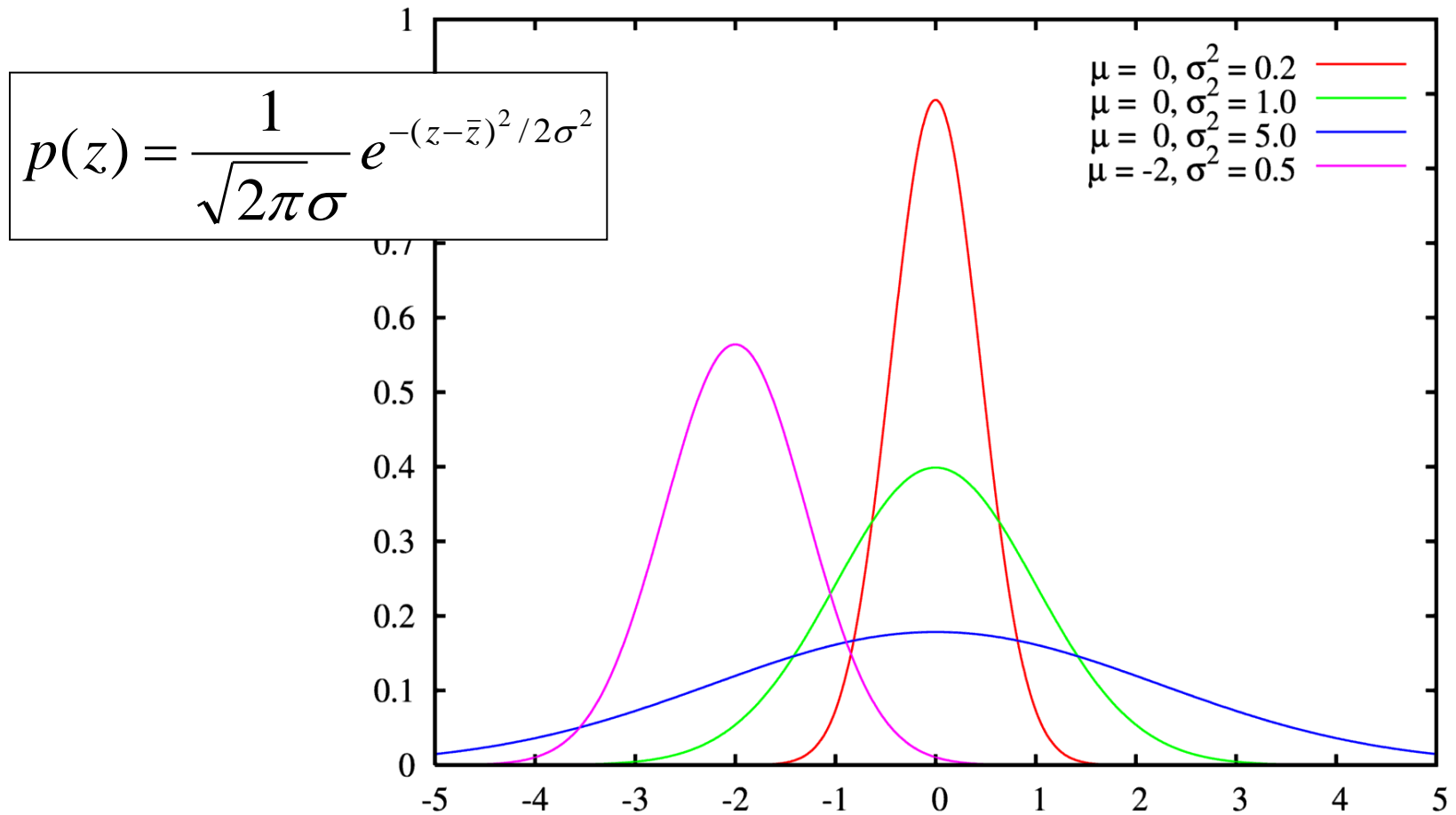
# Model: Gaussian Noise

---

- Gaussian PDF (Probability Density Function).
- Great approximation of reality.
  - Models noise as a sum of various small noise sources, which is indeed what happens in reality.



# Model: Gaussian Noise



---

# Model: Salt and Pepper Noise

---

- Considers that a value can randomly assume the MAX or MIN value for that sensor.
  - Happens in reality due to the malfunction of isolated image sensors.



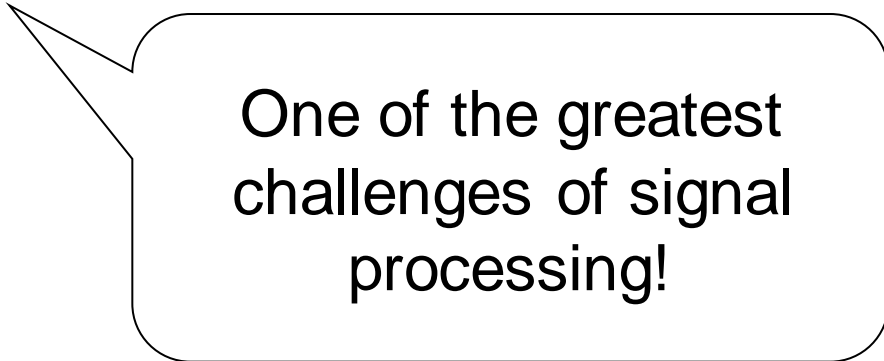
---

# How do we handle it?

---

- **Not always trivial!**

- Frequency filters.
- Estimate the degradation function.
- Inverse filtering.
- ...



One of the greatest challenges of signal processing!

---

# Resources

---

- Gonzalez & Woods – Chapters 5 and 6
- <http://www.howstuffworks.com/>