COMP3421

Radiosity, Colour

Robert Clifton-Everest

Email: robertce@cse.unsw.edu.au

Recap: Global lighting

- In reality, the light falling on a surface comes from everywhere. Light from one surface is reflected onto another surface and then another, and another, and...

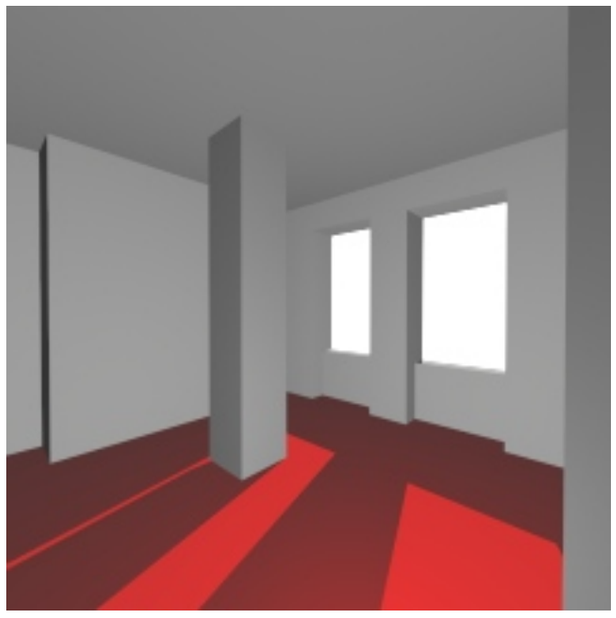
- Methods that take this kind of multi-bounce lighting into account are called global lighting methods.

Raytracing and Radiosity

- There are two main methods for global lighting:
  - Raytracing models specular reflection and refraction.
  - Radiosity models diffuse reflection.
- Both methods are computationally expensive and are rarely suitable for real-time rendering.

Radiosity

- Radiosity is a global illumination technique which performs indirect diffuse lighting.



direct lighting + ambient

global illumination

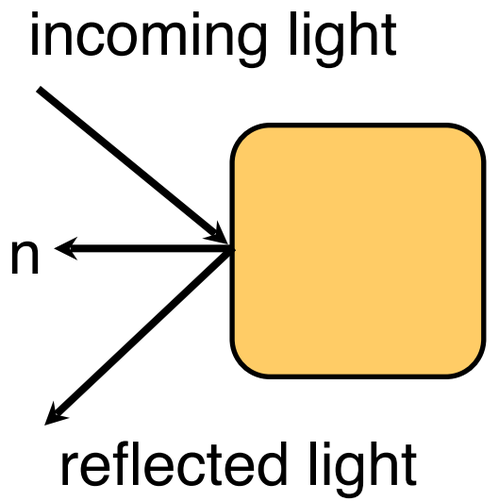
Radiosity

- Direct lighting techniques only take into account light coming directly from a source.

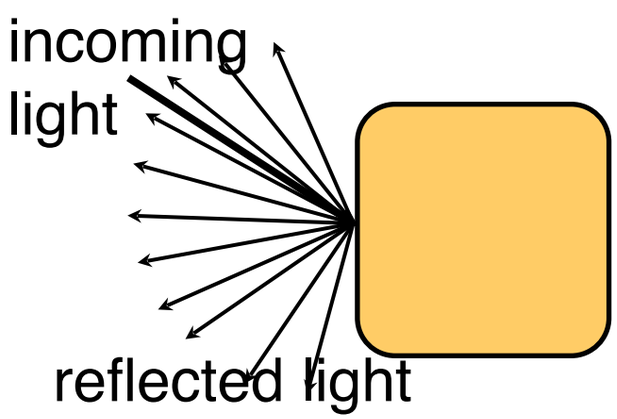
- Raytracing takes into account specular reflections of other objects.

- Radiosity takes into account diffuse reflections of everything else in the scene.

Ray tracing vs Radiosity

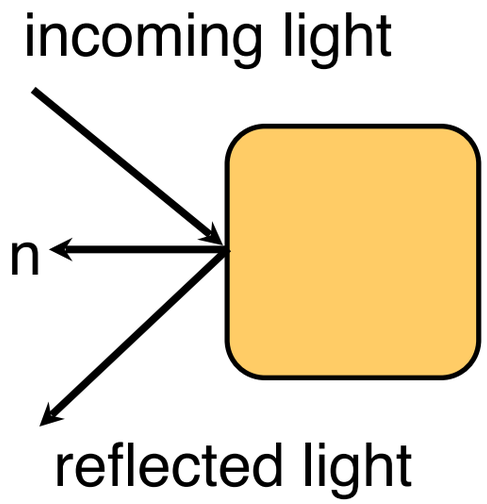


Specular reflection

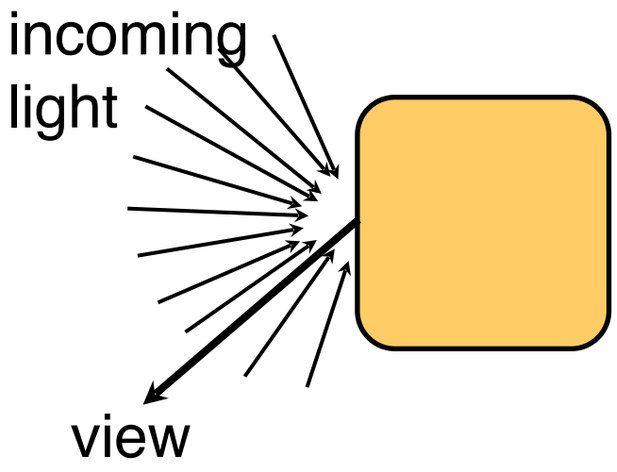


Diffuse reflection

Ray tracing vs Radiosity



Specular reflection



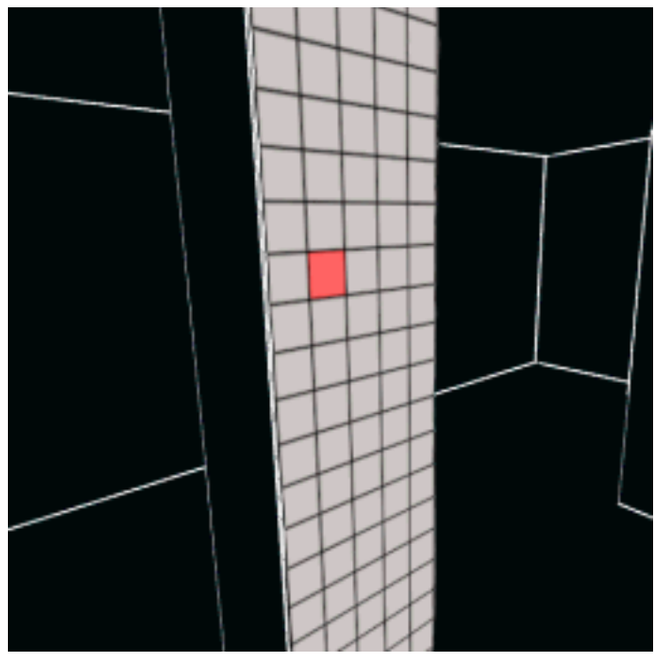
Diffuse reflection

Finite elements

- We can solve the radiosity problem using a finite element method.

- We divide the scene up into small patches.

- We then calculate the energy transfer from each patch to every other patch.



Energy transfer

- The basic equation for energy transfer is:

$$\text{Light output} = \text{Light emitted} + \rho * \text{Light input}$$

- where ρ is the diffuse reflection coefficient.

Energy transfer

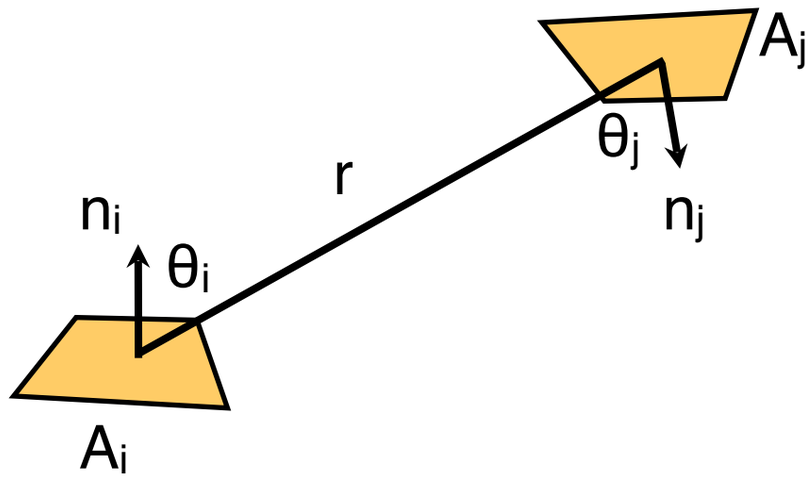
- The light input to a patch is a weighted sum of the light output by every other patch.

$$B_i = E_i + \rho_i \sum_j B_j F_{ij}$$

B_i is the radiosity of patch i
E_i is the energy emitted by patch i
ρ_i is the reflectivity of patch i
F_{ij} is a form factor which encodes what fraction of light from patch j reaches patch i.

Form factors

- The form factors F_{ij} depend on
  - the shapes of patches i and j
  - the distance between the patches
  - the relative orientation of the patches

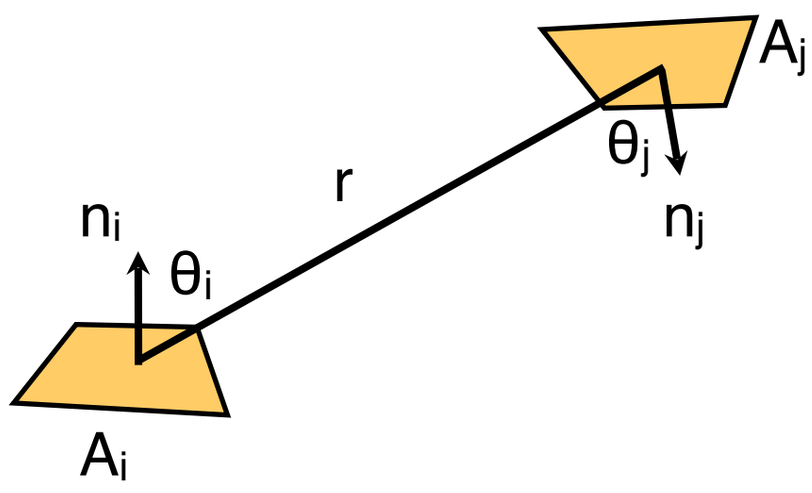


Form factors

Mathematically:

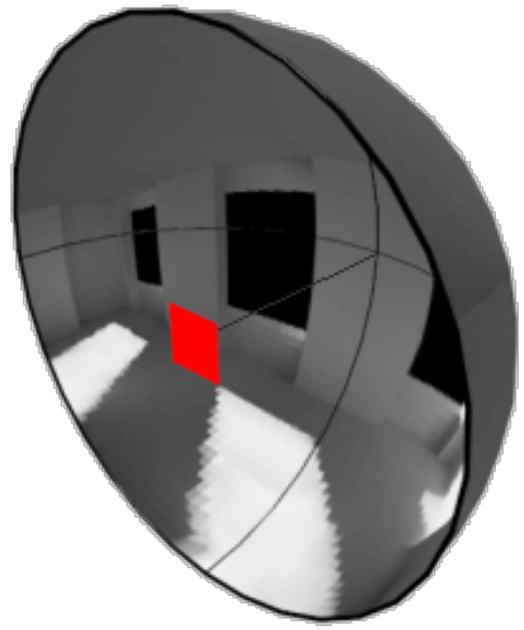
$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos\theta_i \cos\theta_j}{\pi r^2} dA_j dA_i$$

Calculating form factors in this way is difficult and does not take into account occlusion.

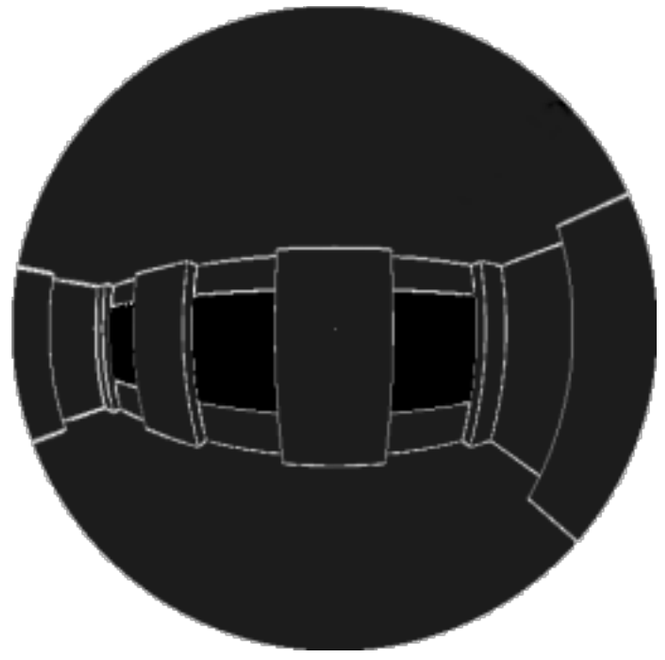


Nusselt Analog

- An easier equivalent approach:

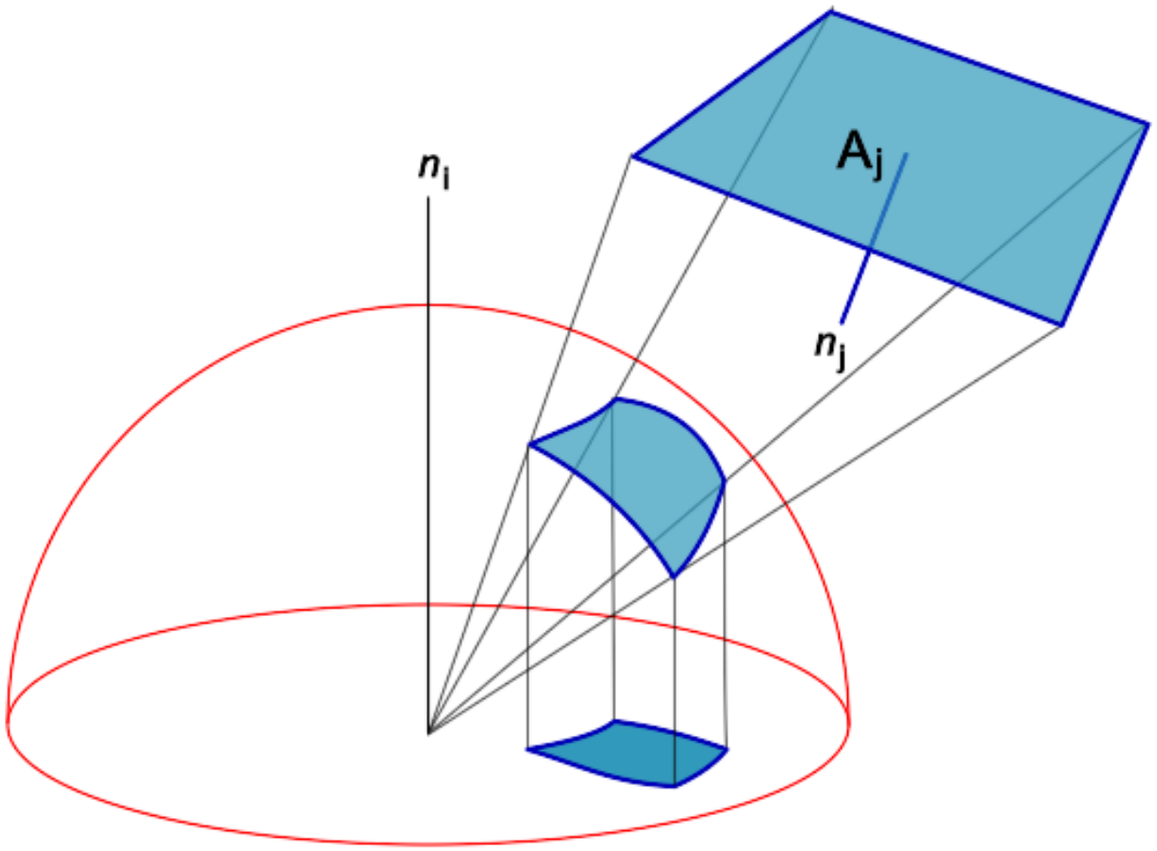


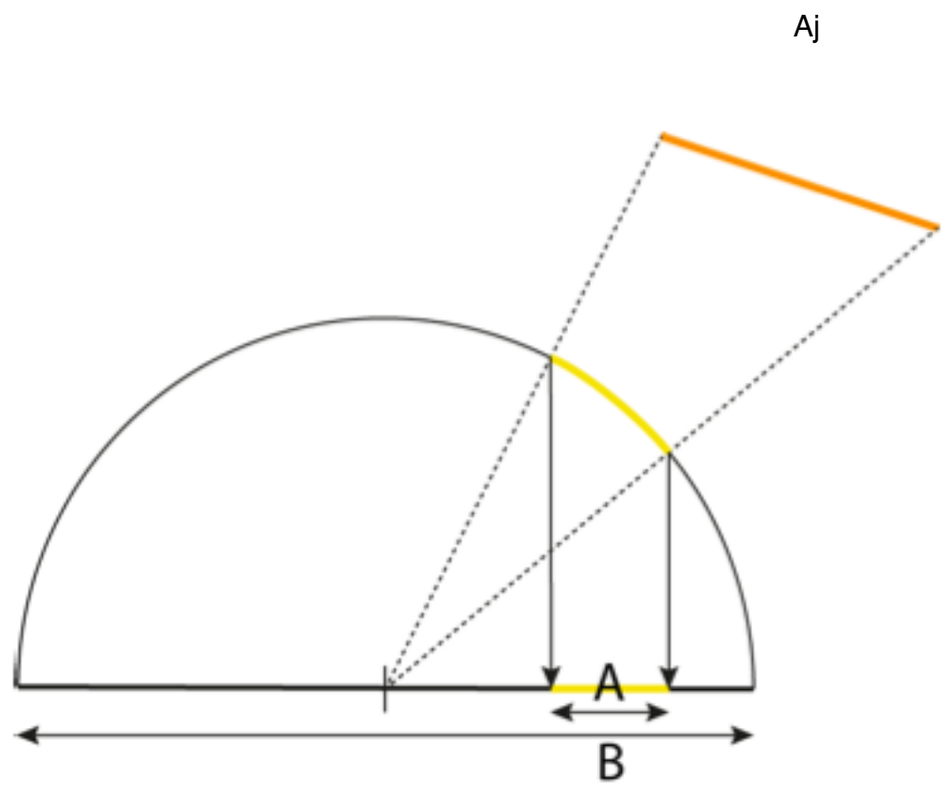
1. render the scene onto a unit hemisphere from the patch's point of view.



2. project the hemisphere orthographically on a unit circle.

3. divide by the area of the circle





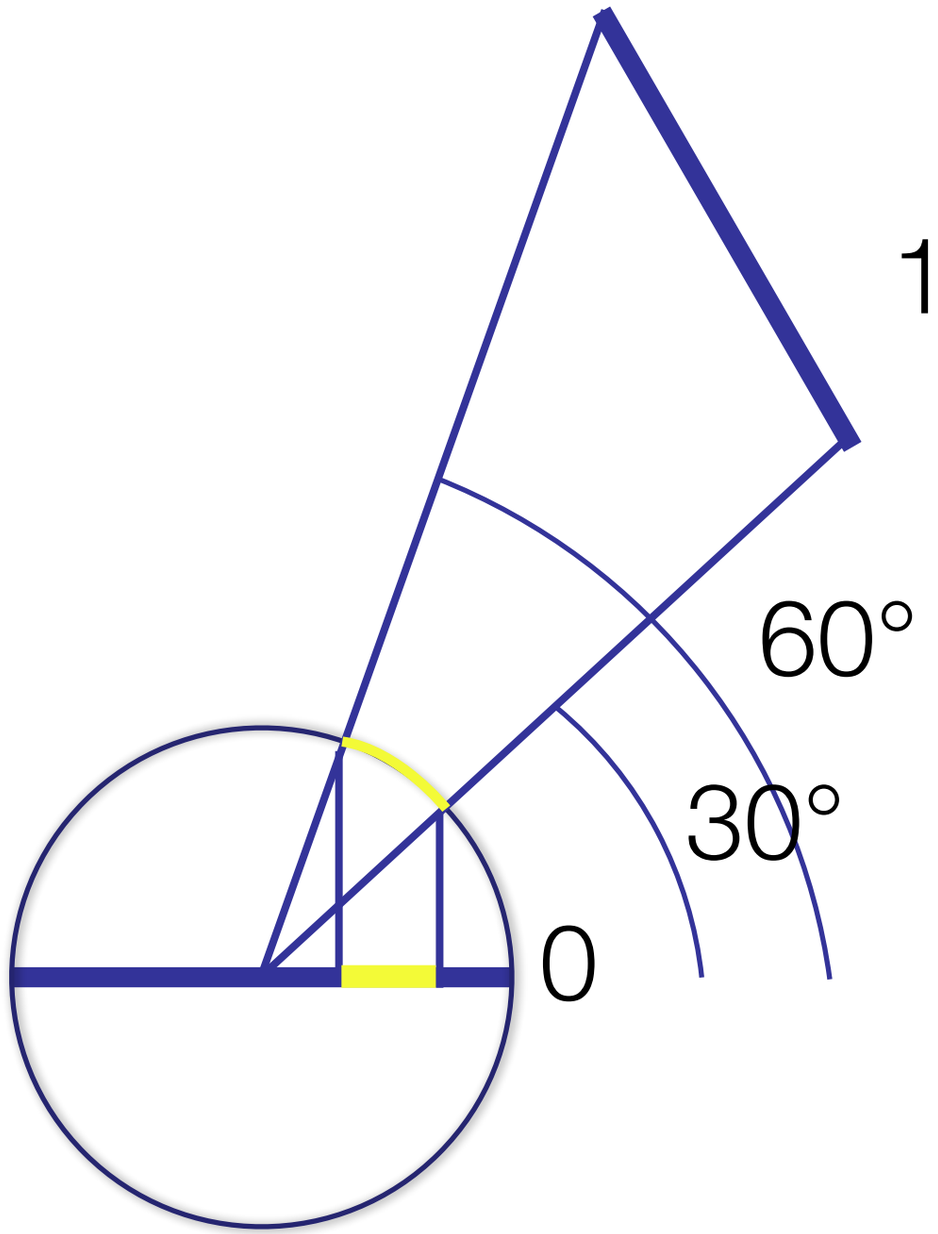
$$F_{ij} = A/B$$

$$\rho = 0.5$$

$$E[0] = 0$$

$$E[1] = 0.8$$

$$F_{01} = F_{10} = (\cos(30) - \cos(60))/2 \approx 0.183$$

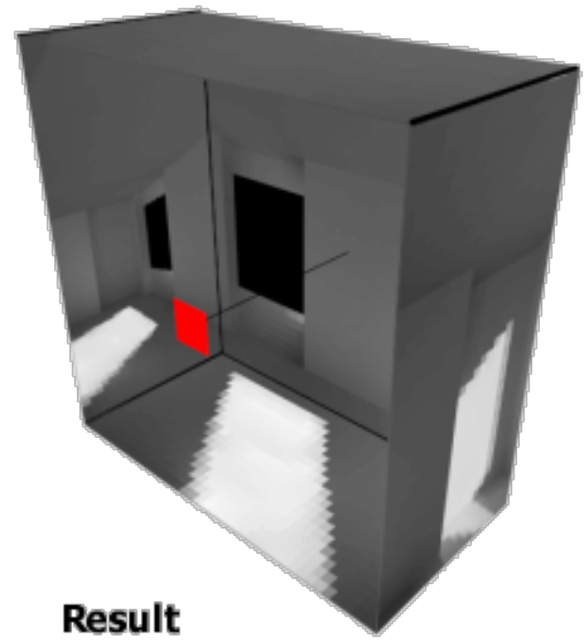


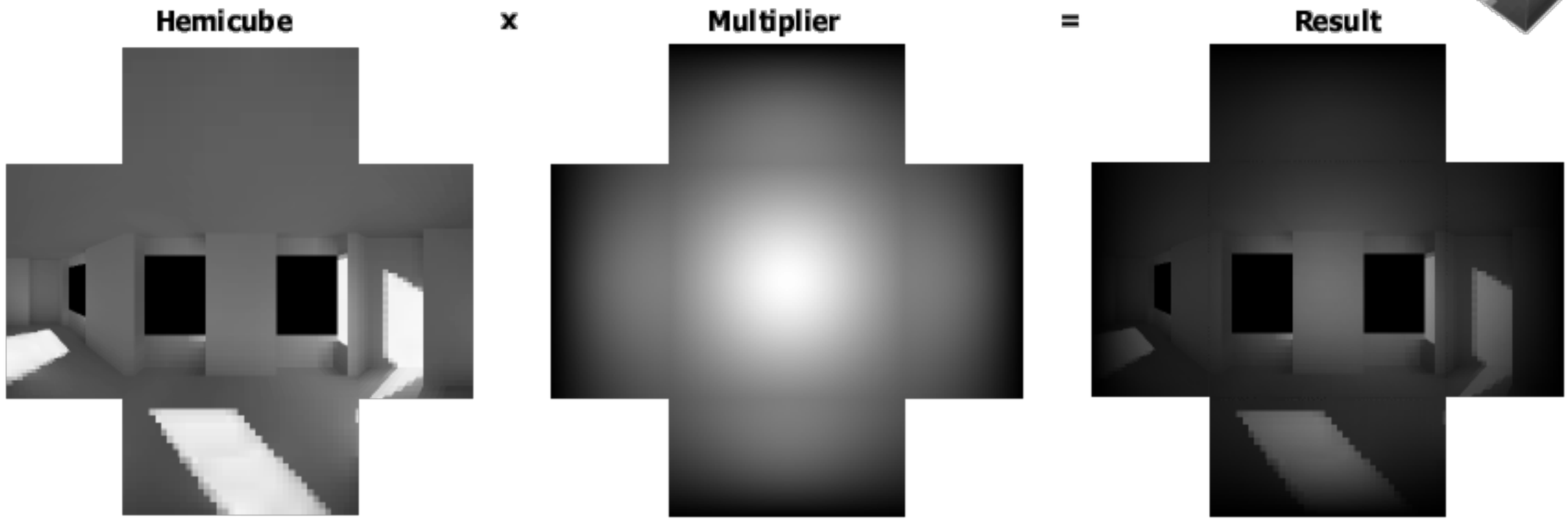
F	0	1
0	0	0.183
1	0.183	0

	B_0	B_1
1	0	0.8
2	0.073	0.8
3	0.073	0.807
4	0.074	0.807

The hemicube method

- A simpler method is to render the scene onto a hemicube and weight the pixels to account for the distortion.





Solving

- We can view the radiosity equation as a system of linear equations

$$B_i = E_i + \rho_i \sum_j B_j F_{ij}$$

$$B_1 = E_1 + \rho_1 (B_1 F_{11} + B_2 F_{12} + \ldots + B_N F_{1N})$$
$$B_2 = E_2 + \rho_2 (B_1 F_{21} + B_2 F_{22} + \ldots + B_N F_{2N})$$
$$\ldots$$
$$B_N = E_N + \rho_N (B_1 F_{N1} + B_2 F_{N2} + \ldots + B_N F_{NN})$$

Solving

- This system of equations can be expressed as a matrix equation:

$$\begin{pmatrix} 1-\rho_1 F_{11} & -\rho_1 F_{12} & \cdots & -\rho_1 F_{1N} \\ -\rho_2 F_{21} & 1-\rho_2 F_{22} & \cdots & -\rho_2 F_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ -\rho_N F_{N1} & -\rho_N F_{N2} & \cdots & 1-\rho F_{NN} \end{pmatrix} \begin{pmatrix} B_1 \\ B_2 \\ \vdots \\ B_N \end{pmatrix} = \begin{pmatrix} E_1 \\ E_2 \\ \vdots \\ E_N \end{pmatrix}$$

- In practice N is very large making exact solutions infeasible.

Iterative approximation

- One simple solution is merely to update the radiosity values in multiple passes:

$$B_i = E_i + \rho_i \sum_j B_j F_{ij}$$

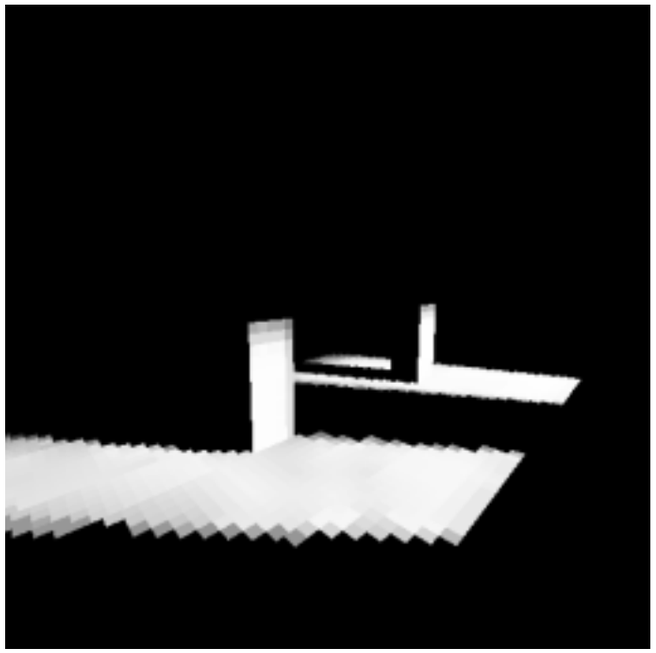
```
for each iteration:
  for each patch i:
    Bnew[i] = E[i]
    for each patch j:
      Bnew[i] +=
        rho[i] * F[i,j] * Bold[j];
  swap Bold and Bnew
```

Iterative approximation

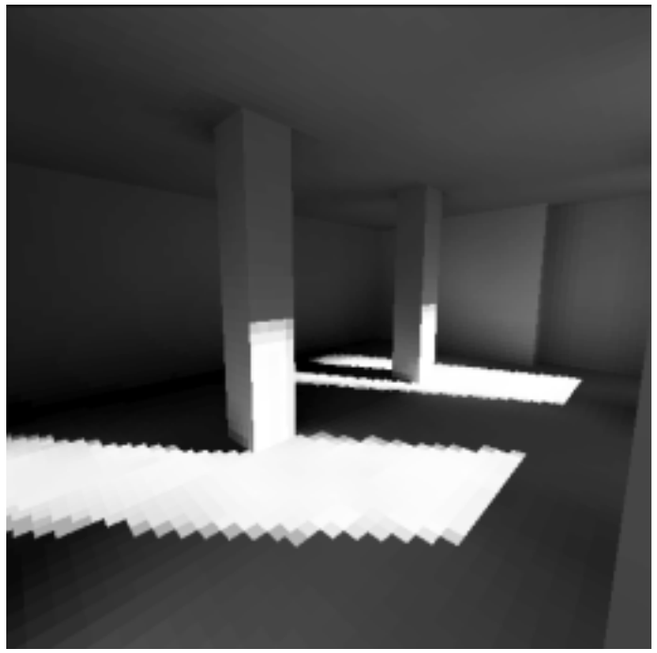
Using direct rendering

```
for each iteration:
   for each patch i:
   Bnew[i] = E[i]
   S = RenderScene(i,Bold)
   B = Sum of pixels in S
   Bnew[i] += rho[i]*B
swap Bold and Bnew
```

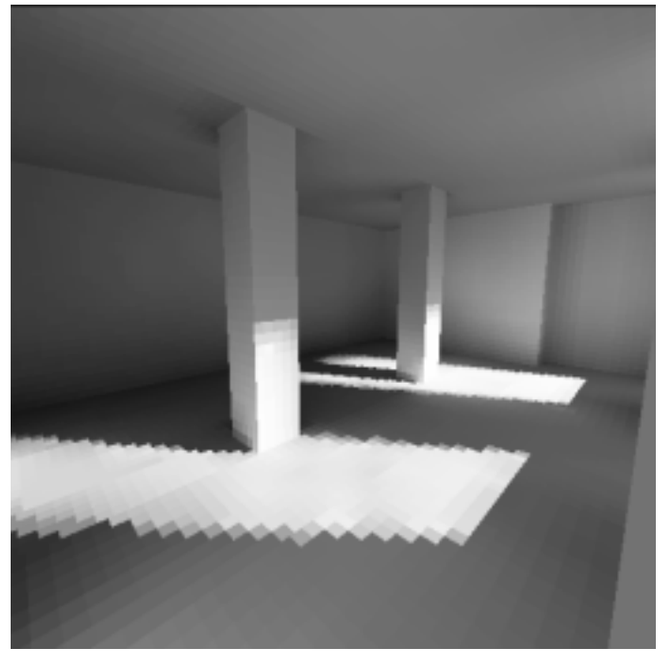
Iterative approximation



first pass
(direct lighting)

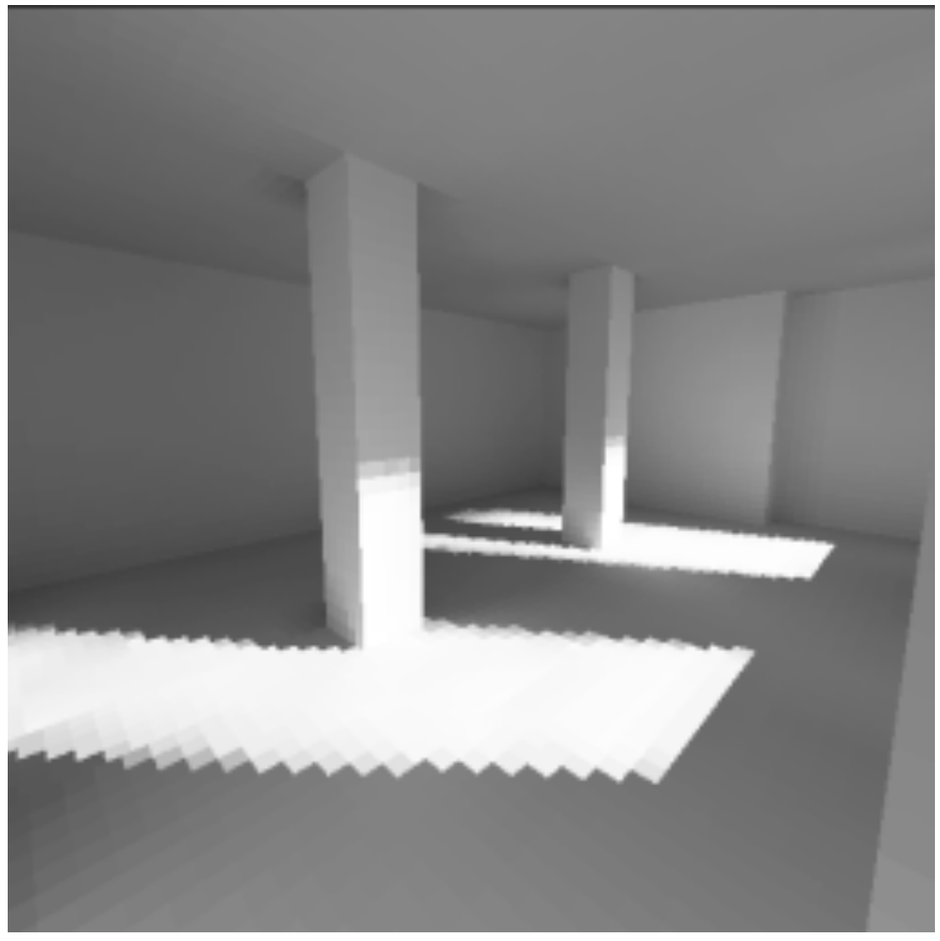


second pass
(one bounce)



third pass
(two bounces)

16th Pass



Progressive refinement

- The iterative approach is inefficient as it spends a lot of time computing inputs from patches that make minimal or no contribution.
- A better approach is to prioritise patches by how much light they output, as these patches will have the greatest contribution to the scene.

Progressive refinement

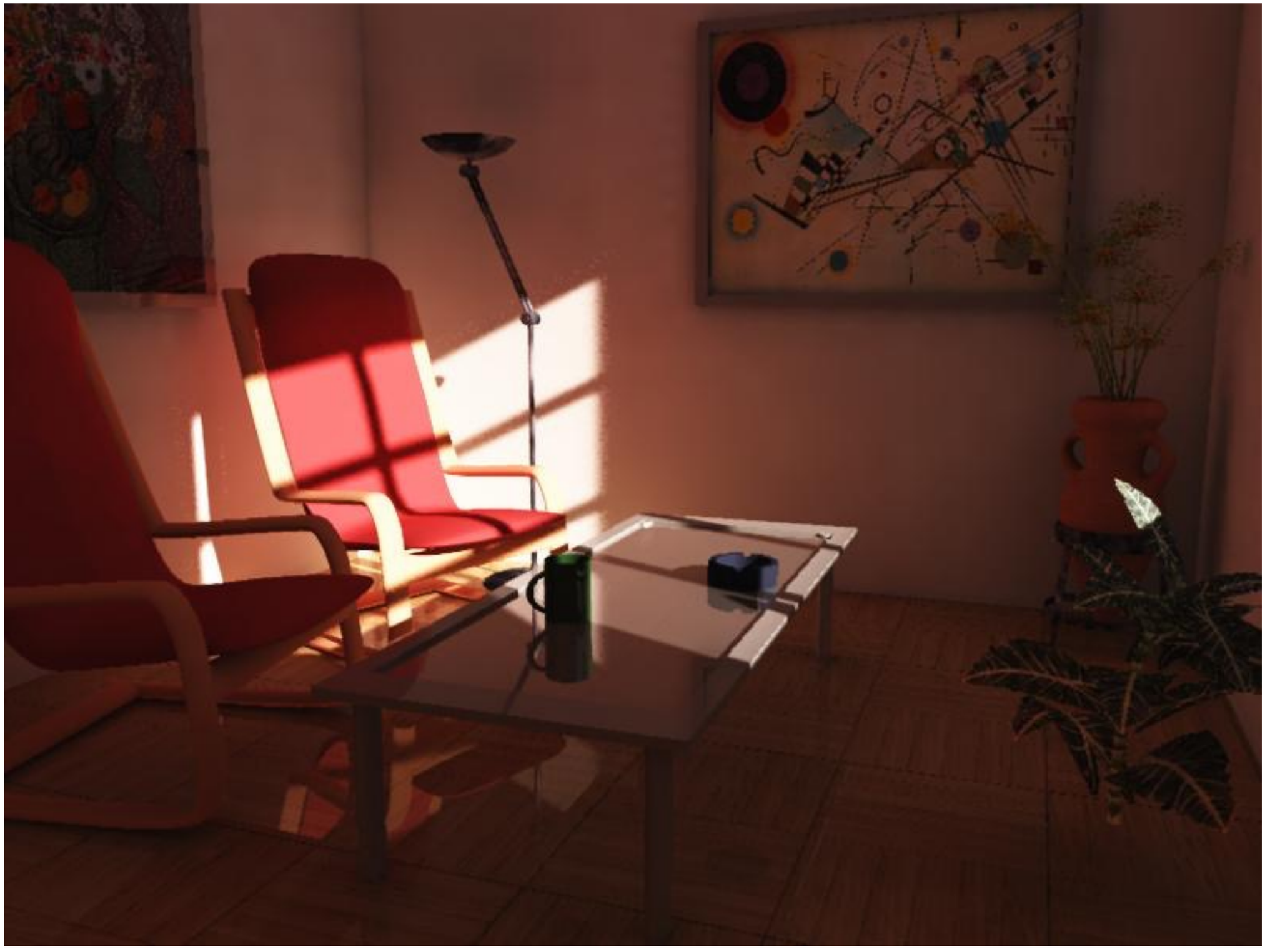
```
for each patch i:
    B[i] = dB[i] = E[i]
iterate:
  select patch i with max dB[i]:
  calculate F[i][j] for all j
  for each patch j:
    dRad = rho[j] * B[i] *
           F[i][j] * A[j] / A[i]
    B[j] += dRad
    dB[j] += dRad
dB[i] = 0
```

In practice

- Radiosity is computationally expensive, so rarely suitable for real-time rendering.

- However, it can be used in conjunction with light mapping.

The payoff



Geometric light sources

Path Tracing

- Uses a Monte Carlo method to accurately approximate global illumination.

- Outside the realm of this course.

- Monte Carlo: https://www.scratchapixel.com/lessons/mathematics-physics-for-computer-graphics/monte-carlo-methods-in-practice/

- Path tracing: https://www.scratchapixel.com/lessons/3d-basic-rendering/global-illumination-path-tracing

Monte Carlo Demos

- https://www.shadertoy.com/view/llXfR8
- https://www.shadertoy.com/view/4sBfW3
- https://www.shadertoy.com/view/4s3cRr

Real-time Global Illumination

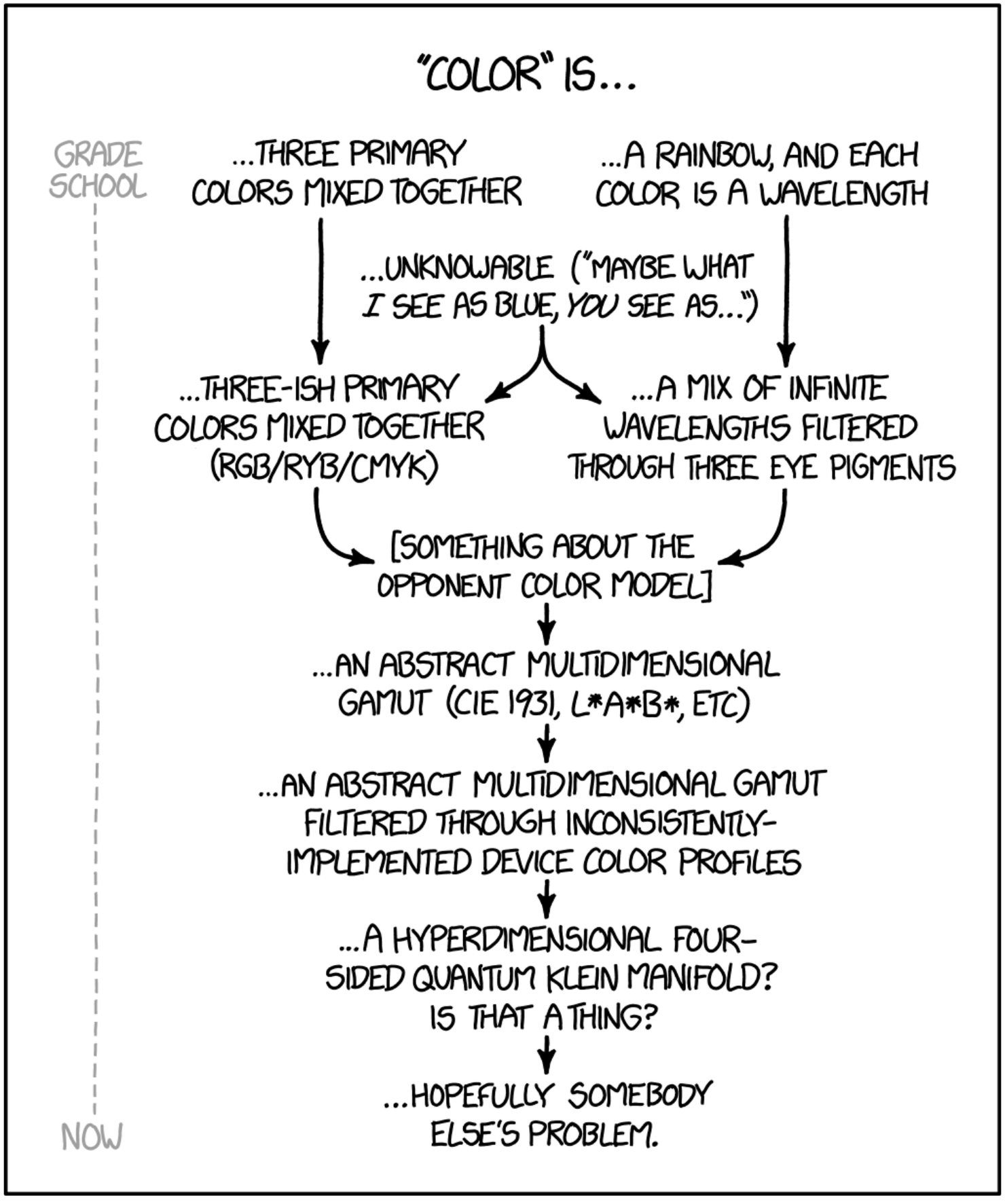
- https://www.shadertoy.com/view/XdtSRn
- https://www.youtube.com/watch?v=BmOppCTh1nA

Sources

- http://freespace.virgin.net/hugo.elias/radiosity/radiosity.htm
- http://www.cs.uu.nl/docs/vakken/gr/2011/gr_lectures.html
- http://www.siggraph.org/education/materials/HyperGraph/radiosity/overview_2.htm
- http://http.developer.nvidia.com/GPUGems2/gpugems2_chapter39.html

Colour

EVOLUTION OF MY UNDERSTANDING OF COLOR OVER TIME:

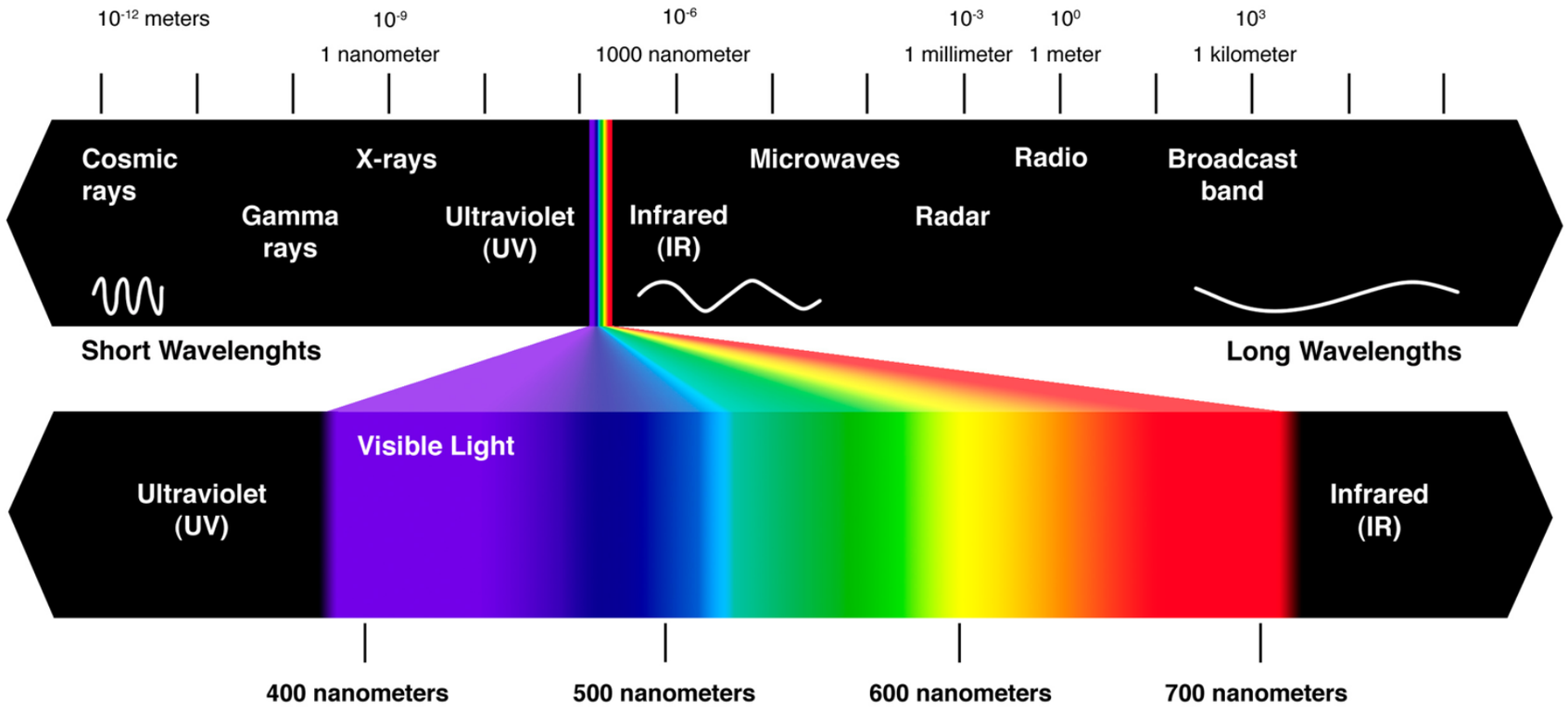


What is colour?

- The experience of colour is complex, involving:
  - Physics of light,
    - Electromagnetic radiation
  - Biology of the eye,
  - Neuropsychology of the visual system.

Physics of light

- Light is an electromagnetic wave, the same as radio waves, microwaves, X-rays, etc.
- The visible spectrum (for humans) consists of waves with wavelength between 400 and 700 nanometers.



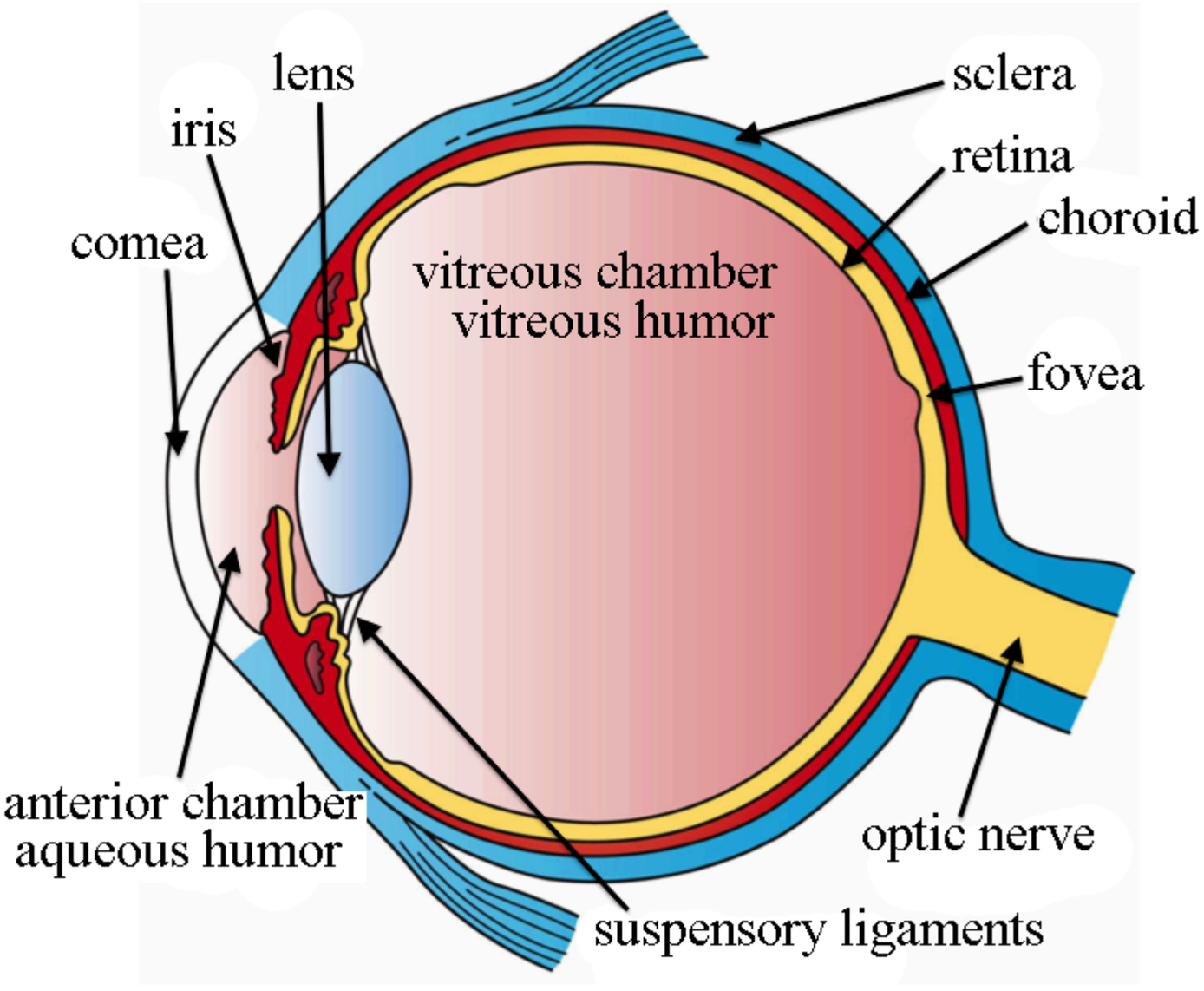
Non-spectral colours

Some light sources, such as lasers, emit light of essentially a single wavelength or “pure spectral” light (red, violet and colors of the rainbow).

Other colours (e.g. white, purple, pink,brown) are non-spectral.

There is no single wavelength for these colours, rather they are mixtures of light of different wavelengths.

The Eye



http://open.umich.edu/education/med/resources/second-look-series/materials

Colour perception

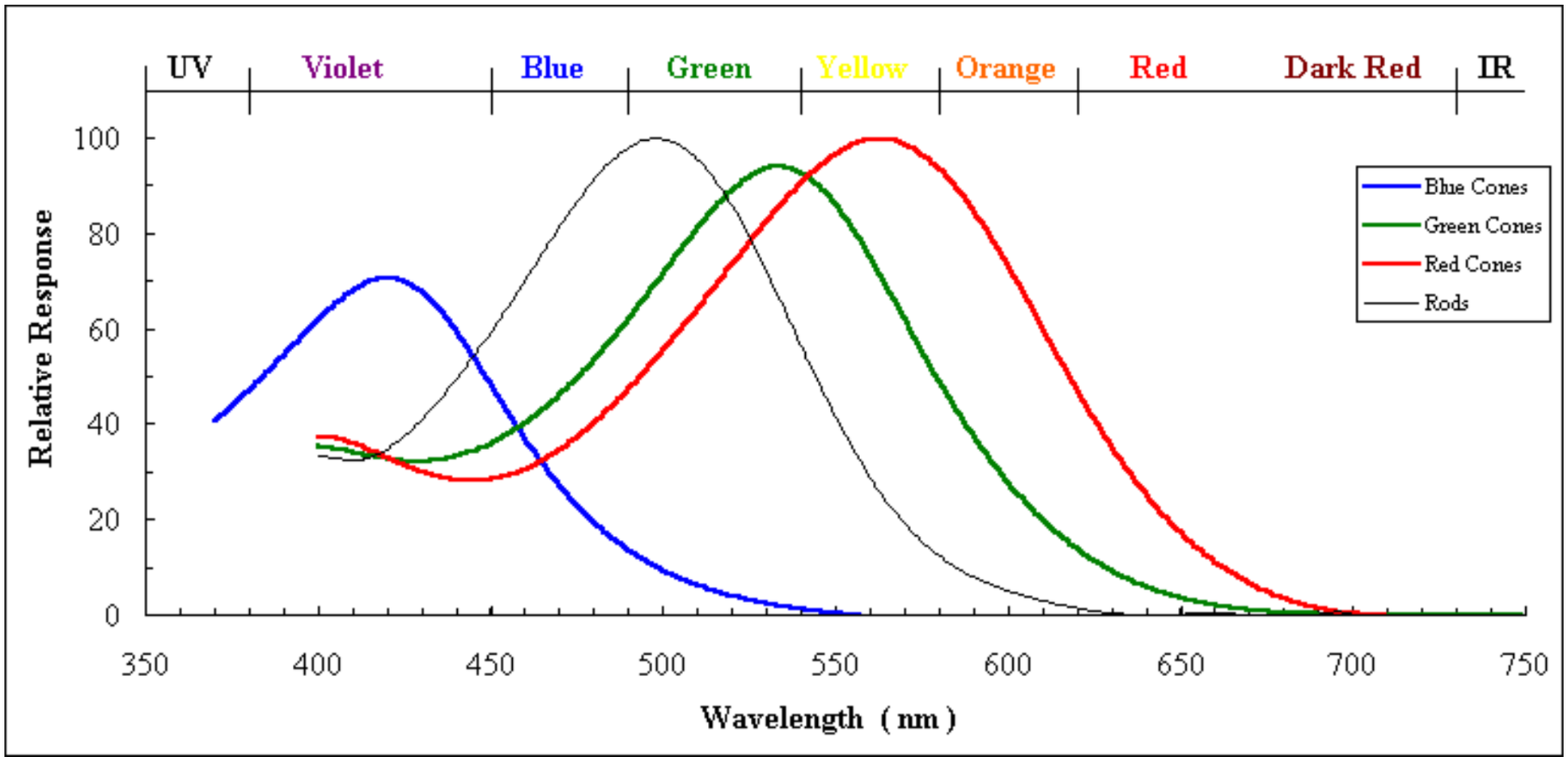
- The retina (back of the eye) has two different kinds of photoreceptor cells: rods and cones.

- Rods are good at handling low-level lighting (e.g. moonlight). They do not detect different colours and are poor at distinguishing detail.

- Cones respond better in brighter light levels. They are better at discerning detail and colour.

Tristimulus Theory

- Most people have three different kinds of cones which are sensitive to different wavelengths.



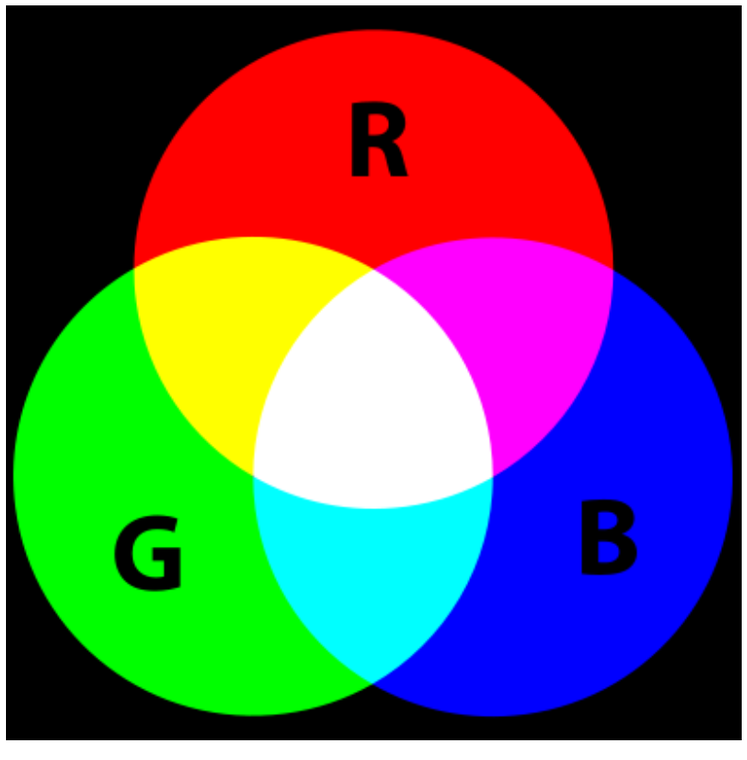
Colour blending

- As a result of this, different mixtures of light will appear to have the same colour, because they stimulate the cones in the same way.

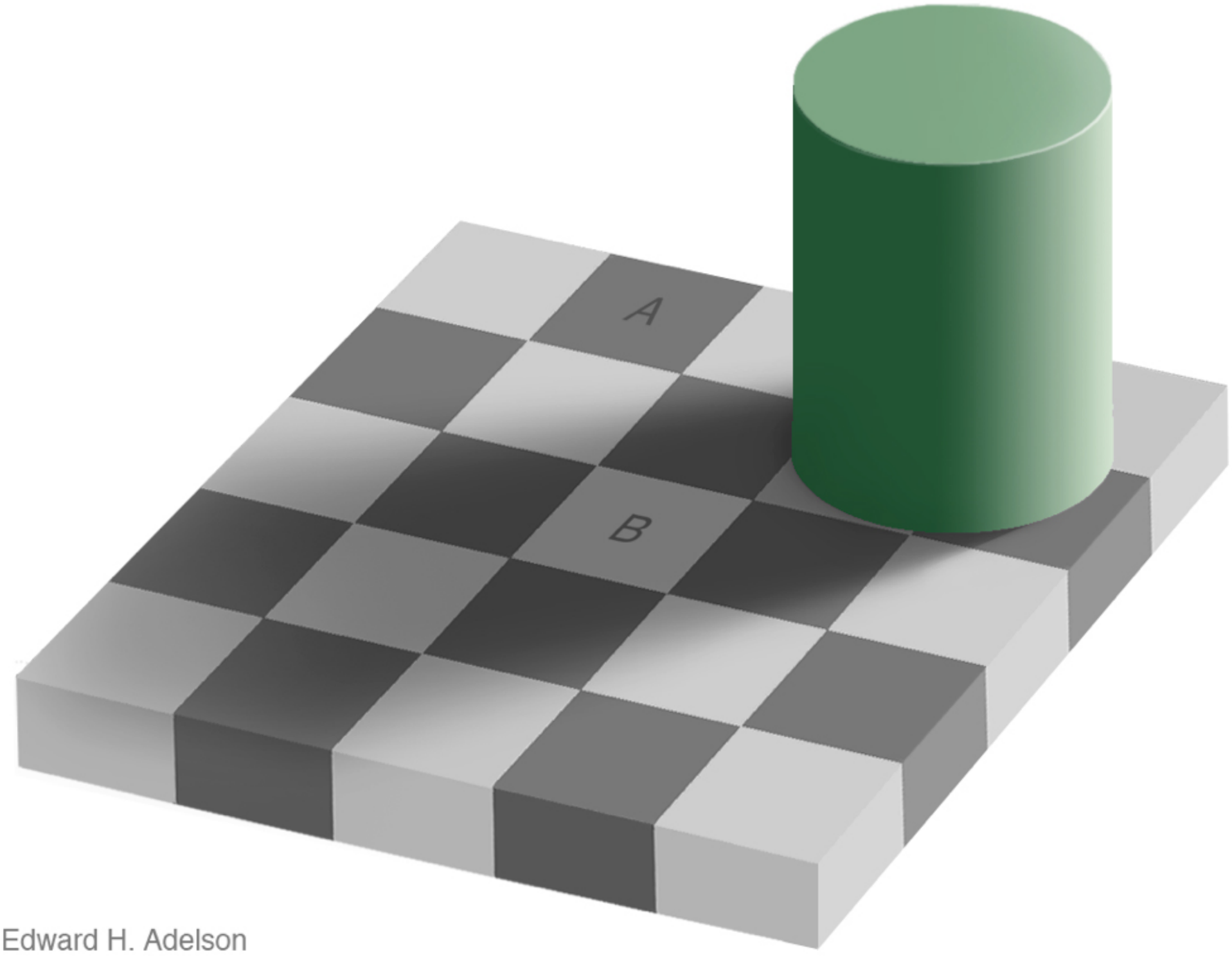
- For example, a mixture of red and green light will appear to be yellow.

Colour blending

- We can take advantage of this in a computer by having monitors with only red, blue and green phosphors in pixels.
- Other colours are made by mixing these lights together.

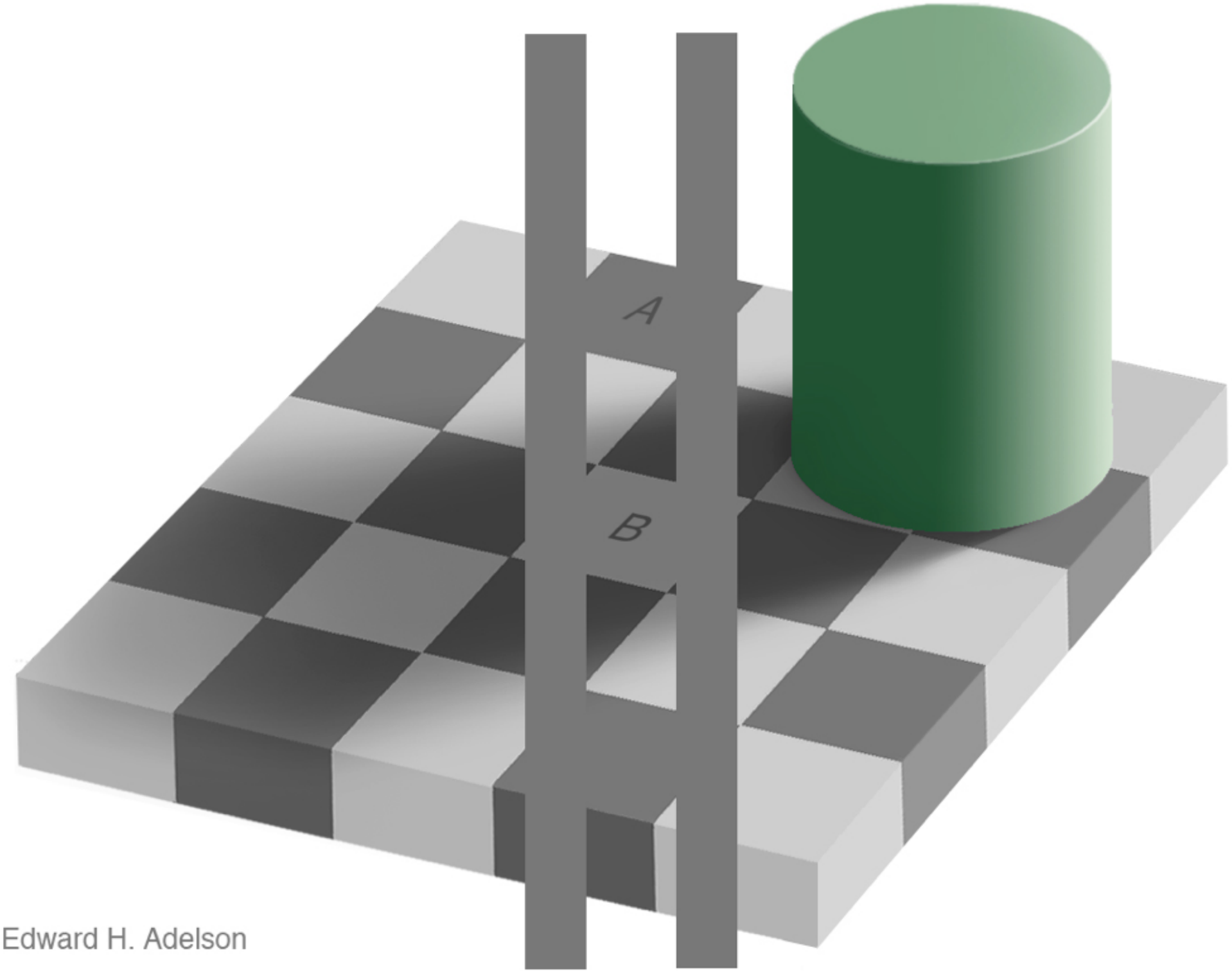


Checker Shadow Illusion



Edward H. Adelson

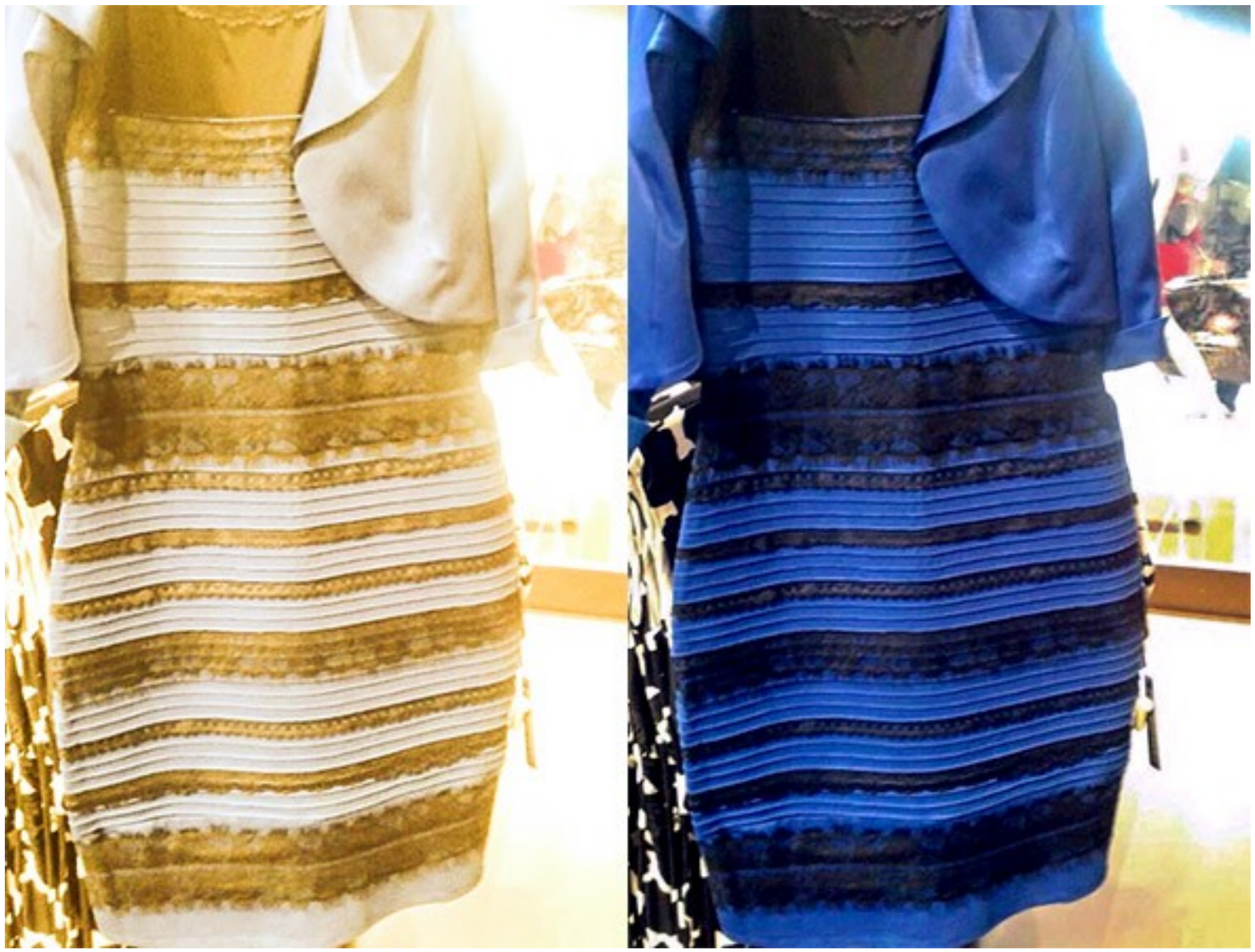
Checker Shadow Illusion

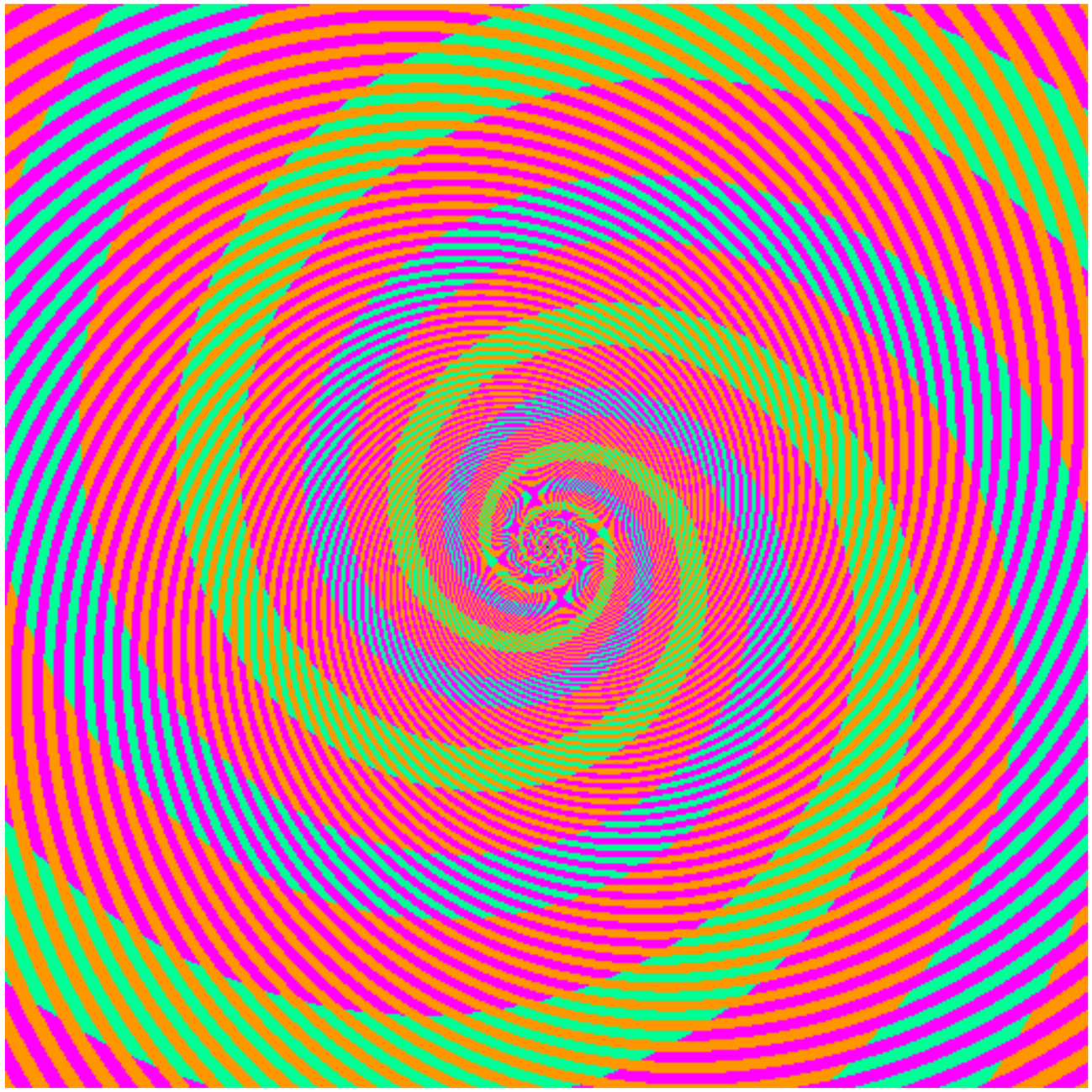


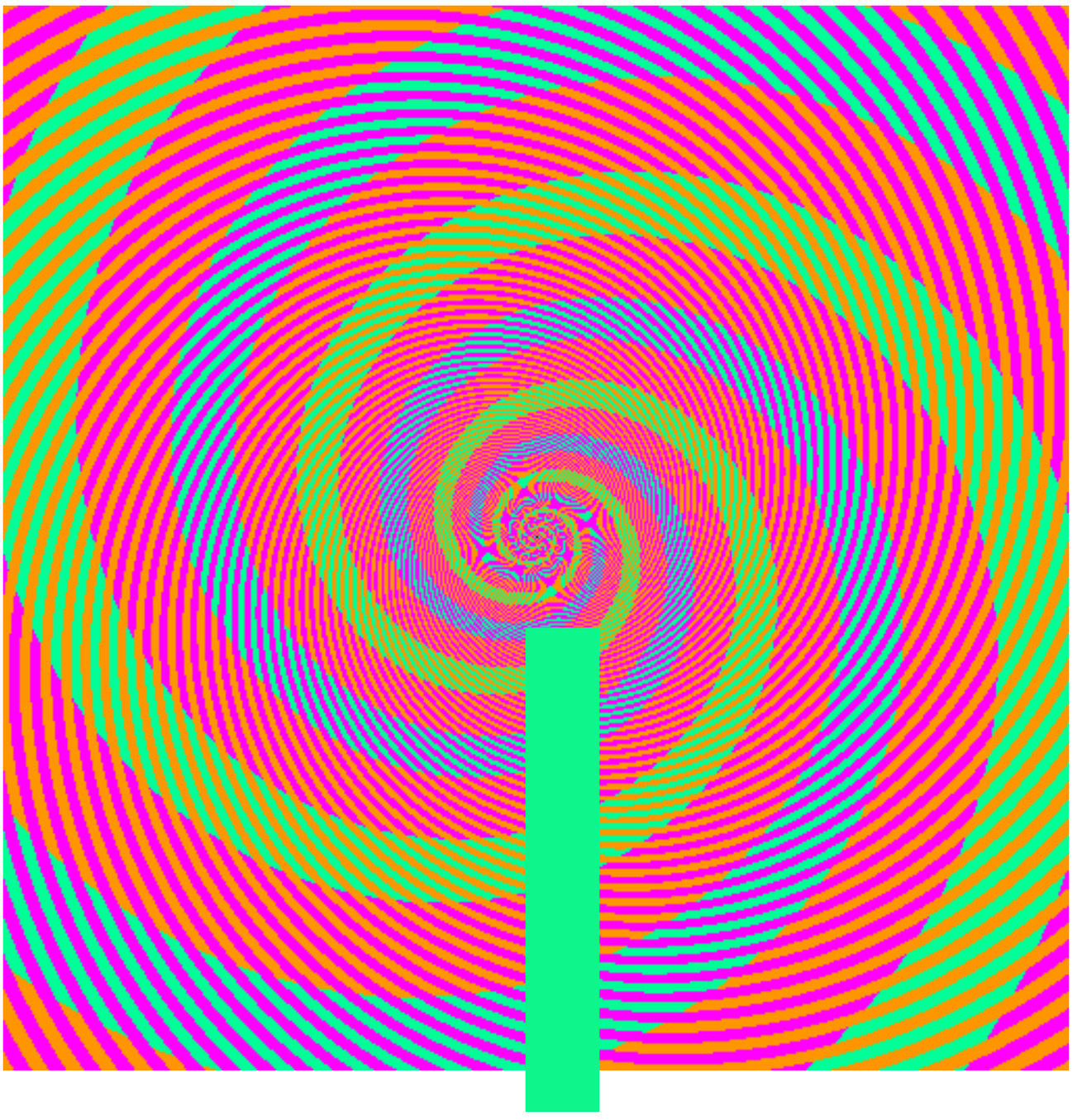
Edward H. Adelson

Color Illusions

Color Illusions



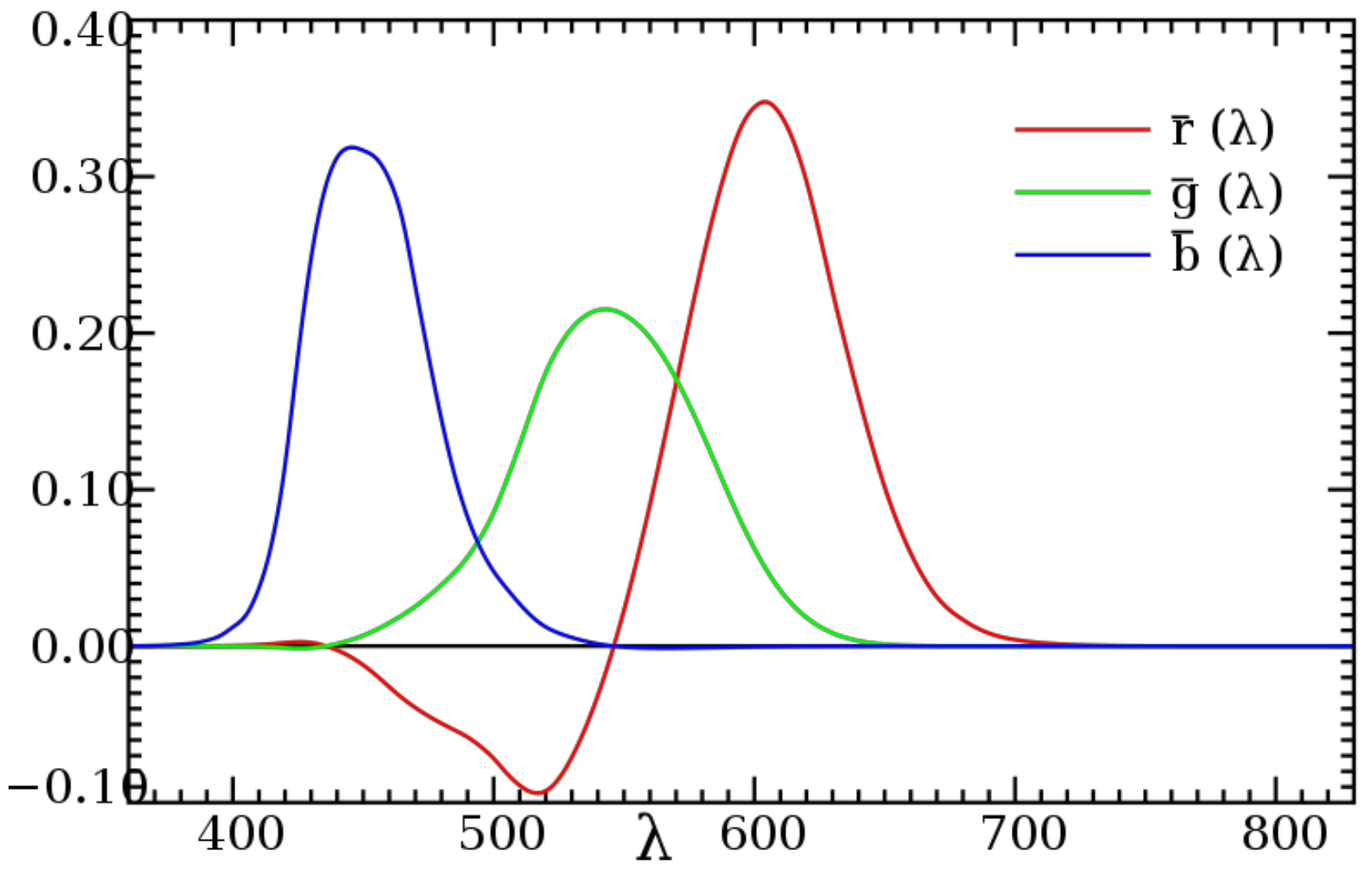




Colour blending

Can we make all colours by adding primaries?

No. Some colours require a negative amount of one of the primaries (typically red).



Colour blending

What does this mean?

Algebraically, we write:

$$C = rR + gG + bB$$

to indicate that colour C is equivalent (appears the same as) *r* units of red, *g* units of green and *b* units of blue.

Colour blending

A colour with wavelength 500nm (cyan/teal) has:

$$C = -0.30R + 0.49G + 0.11B$$

We can rearrange this as:

$$C + 0.30R = 0.49G + 0.11B$$

So if we add 0.3 units of red to colour C, it will look the same as the given combination of green and blue.

Tristimulus Theory and Colour Blending

https://graphics.stanford.edu/courses/cs178/applets/locus.html

https://graphics.stanford.edu/courses/cs178/applets/colormatching.html

Complementary Colors

Colours that add to give white (or at least grey) are called **complementary colours**

eg red and cyan

Retinal fatigue causes complementary colours to be seen in after-images

http://www.animations.physics.unsw.edu.au/jw/light/complementary-colours.htm

Describing colour

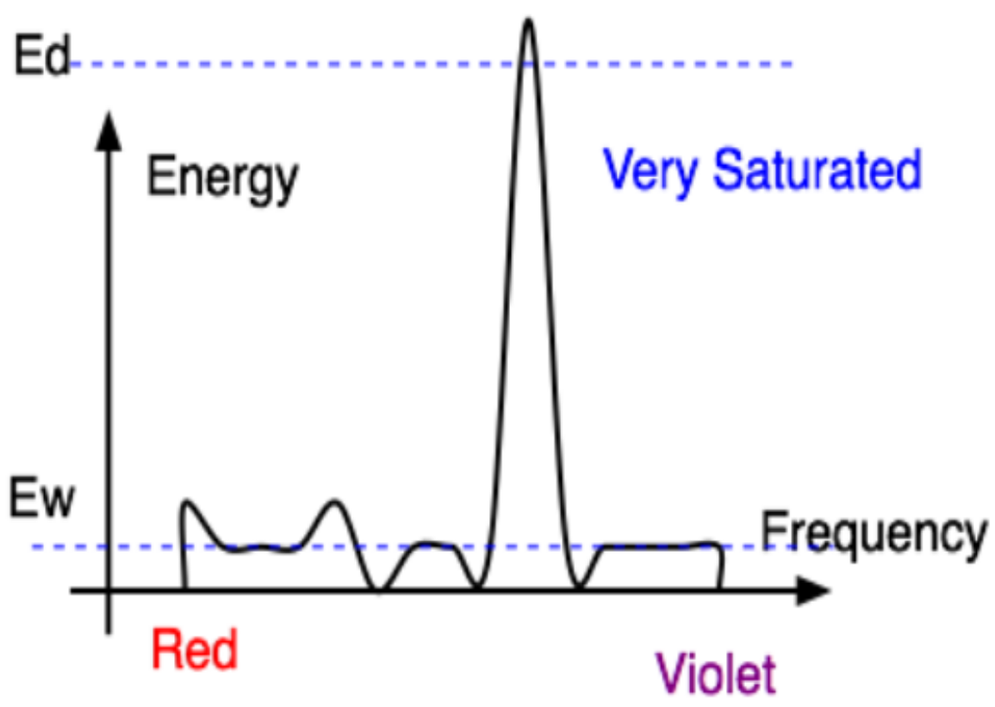
We can describe a colour in terms of its:

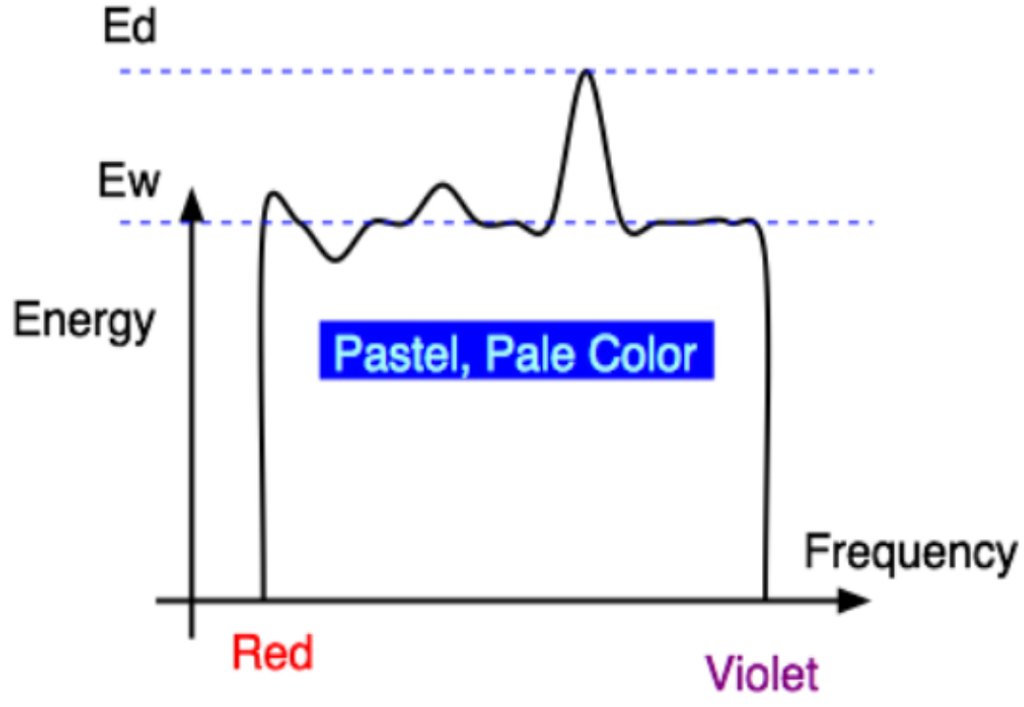
- Hue - the colour of the dominant wavelength such as red
- Luminance - the total power of the light (related to brightness)
- Saturation - the purity of the light
  i.e the percentage of the luminance given by the dominant hue (the more grey it is the more unsaturated)

Spectral Density

Hue is the peak or dominant wavelength

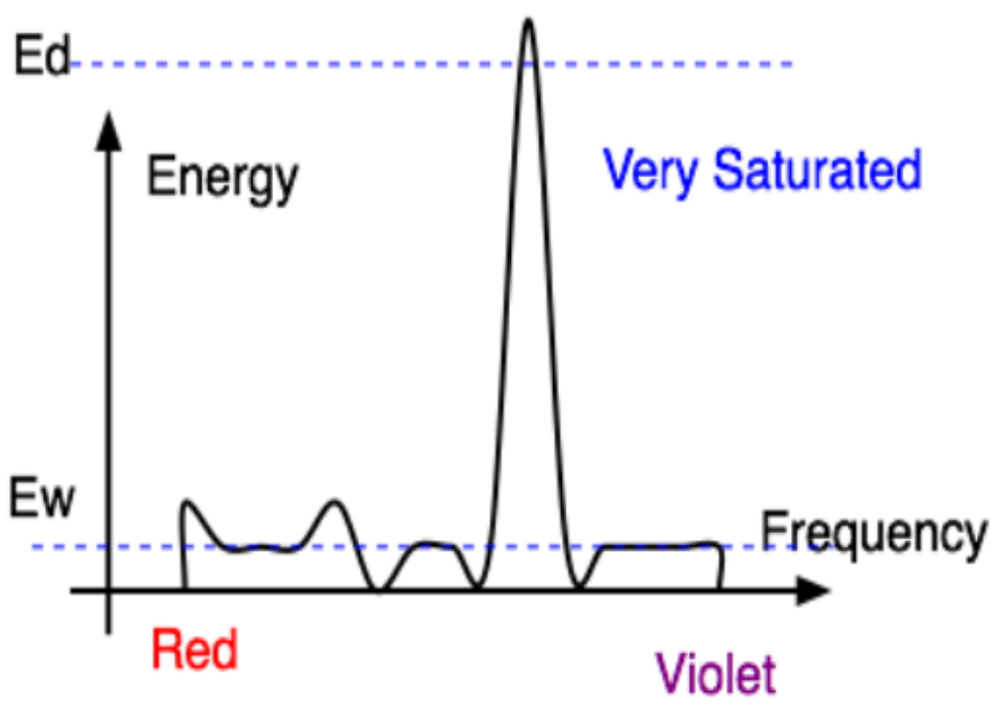
Luminance is related to the intensity or area under the entire spectrum

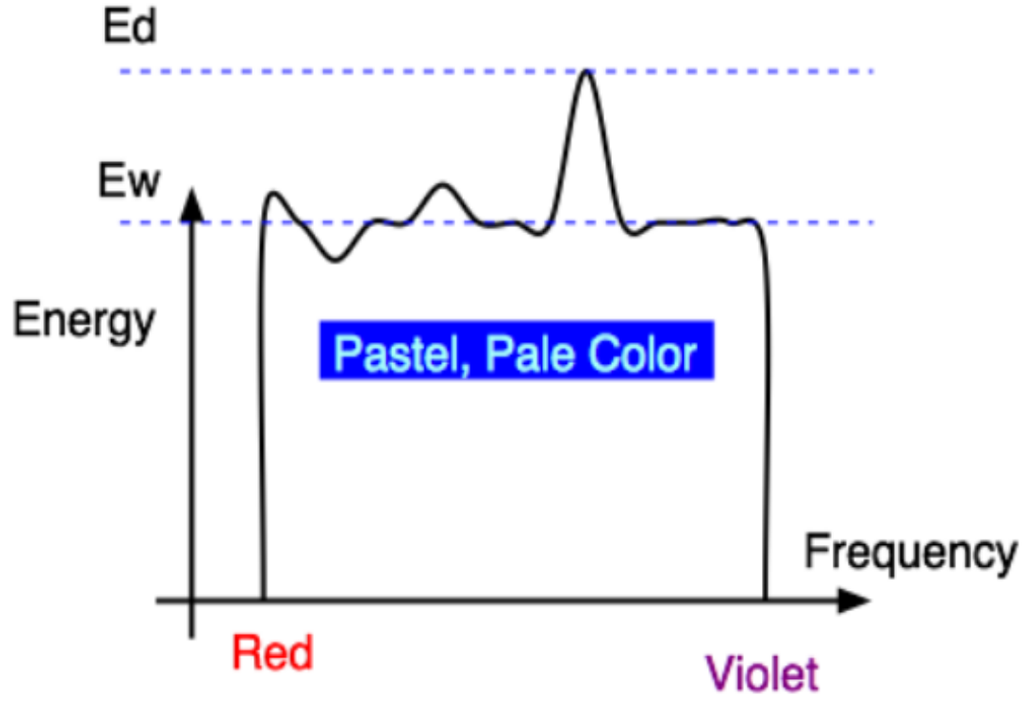




Spectral Density

Saturation is the percentage of intensity in the dominant area





Physics vs Perception

We need to be careful with our language. Physical and perceptual descriptions of light differ.

A red light and a blue light of the same physical intensity will not have the same perceived brightness (the red will appear brighter).

Intensity, Power = physical properties
Luminance, Brightness = perceptual properties

Standardisation

A problem with describing colours as RGB values is that depends on what wavelengths we define as red, green and blue.

Different displays emit different frequencies, which means the same RGB value will result in slightly different colours.

We need a standard that is independent of the particular display.

The CIE standard

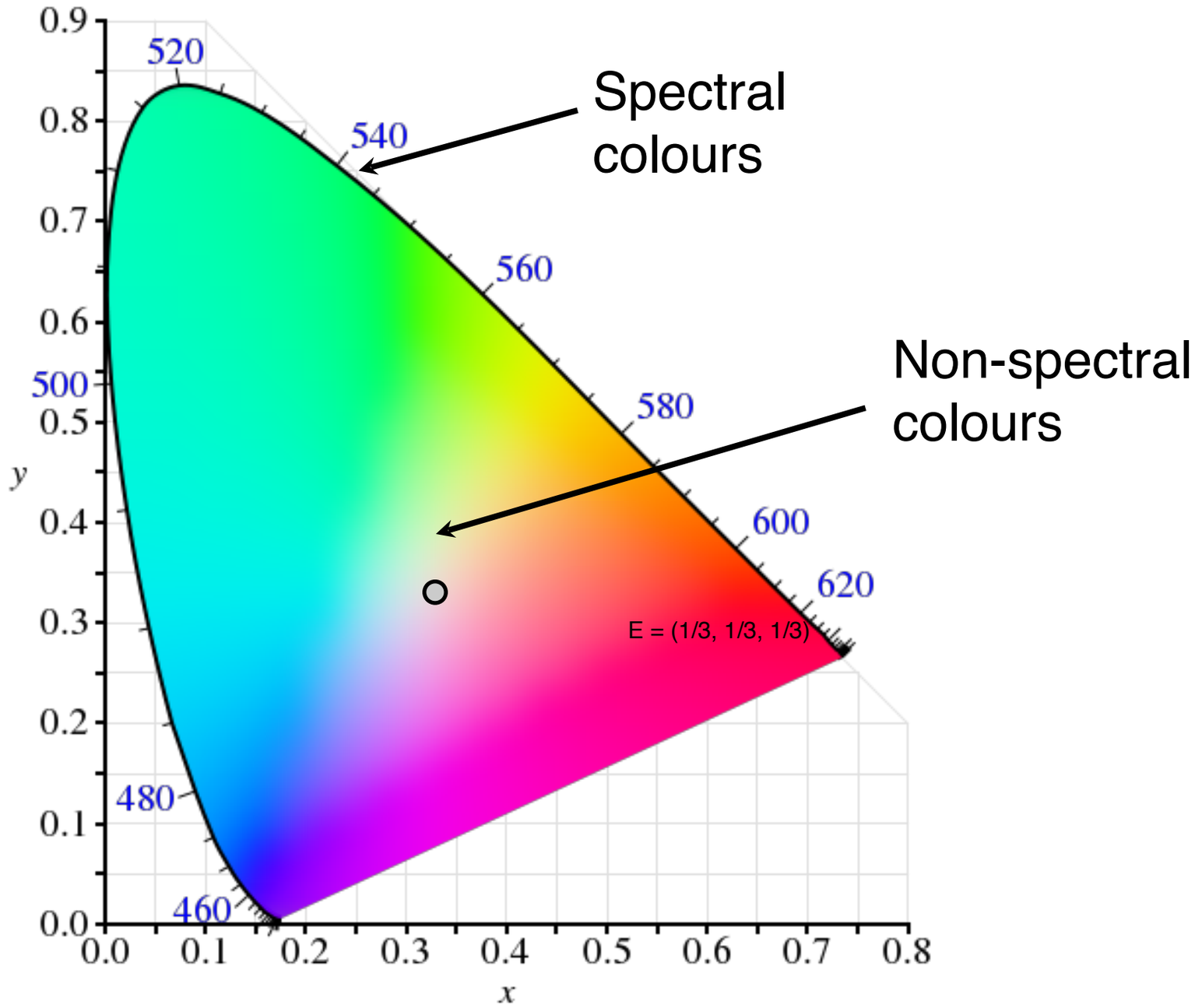
The CIE standard, also known as the XYZ model, is a way of describing colours as a three dimensional vector (x,y,z) with:

$$0 \leq x, y, z \leq 1$$

$$x + y + z = 1$$

X,Y and Z are called imaginary colours.
It is impossible to create pure X, it is just a useful mathematical representation.

The CIE standard



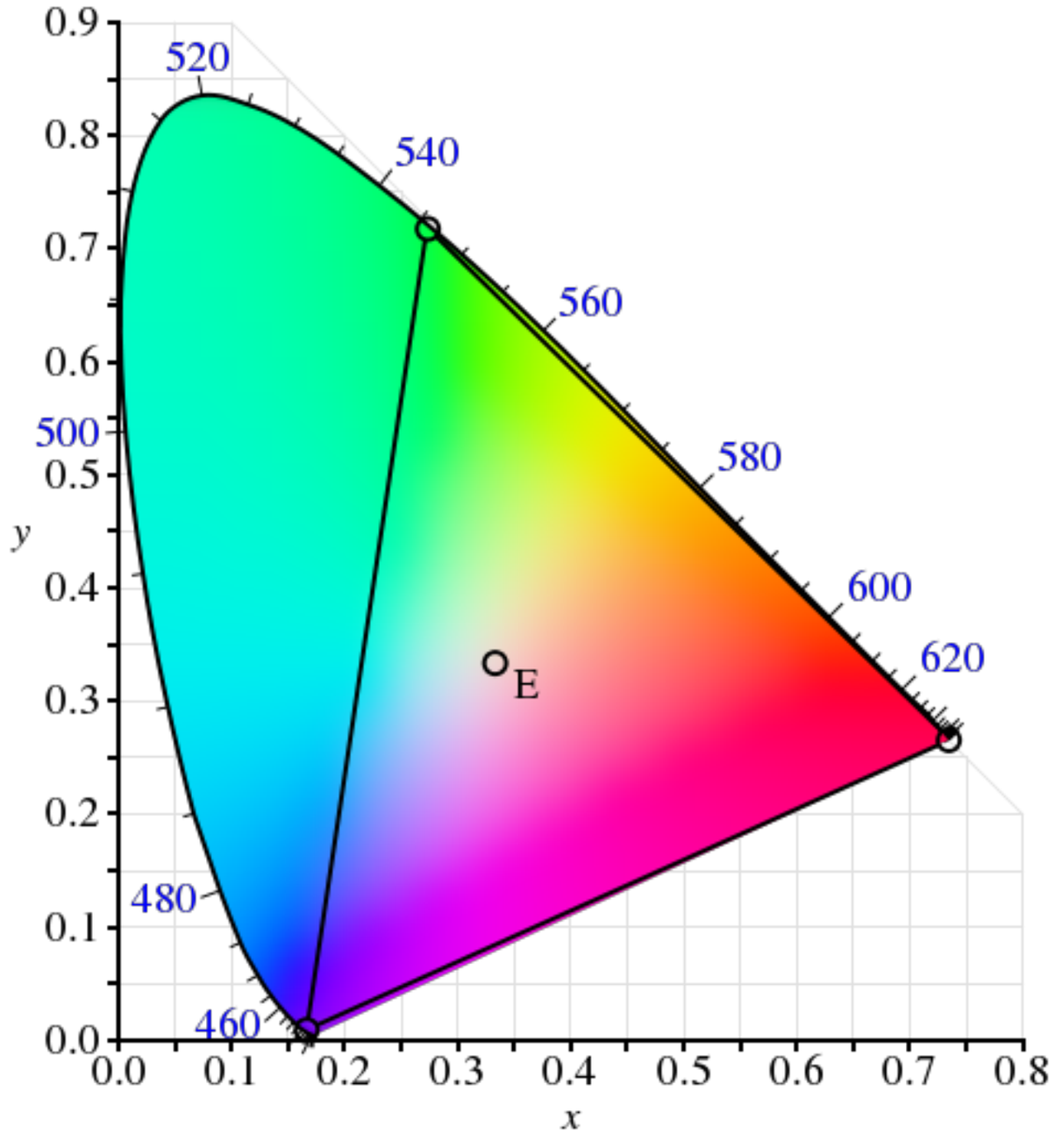
Gamut

Any output device has a certain range of colours it can represent, which we call its gamut.

We can depict this as an area on the CIE chart.

If a monitor has red, green and blue phosphors then the gamut is the interior of the triangle joining those points.

RGB Gamut


0.9
0.8
0.7
0.6
500
0.5
0.4
0.3
0.2
0.1
0.0
y
520
540
560
580
600
620
480
460
E
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
x

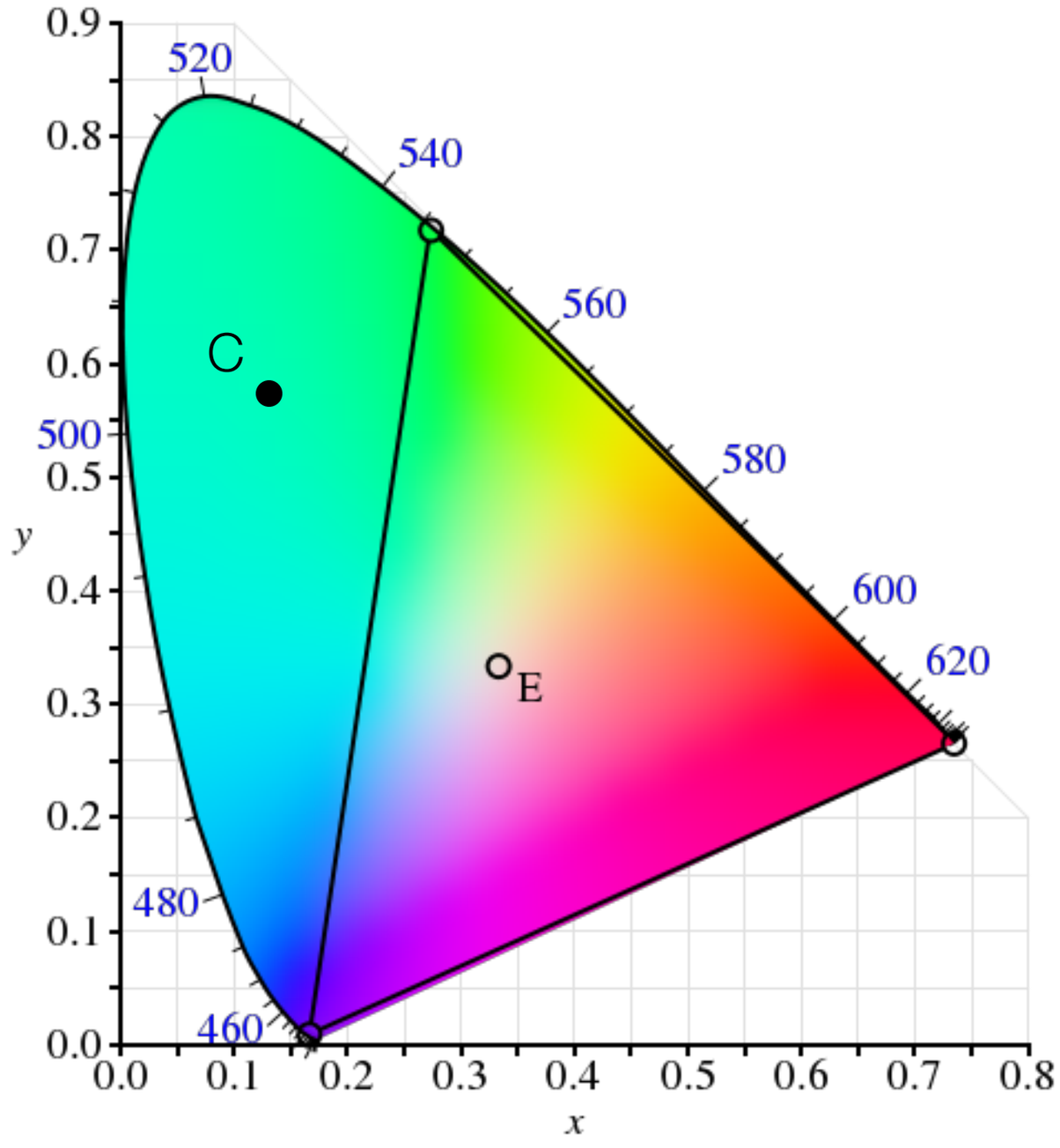
Gamut mapping

How do we map an (x, y, z) colour from outside the gamut to a colour we can display?

We want to maintain:

- Approximately the same hue
- Relative saturation to other colours in the image.

Rendering intents

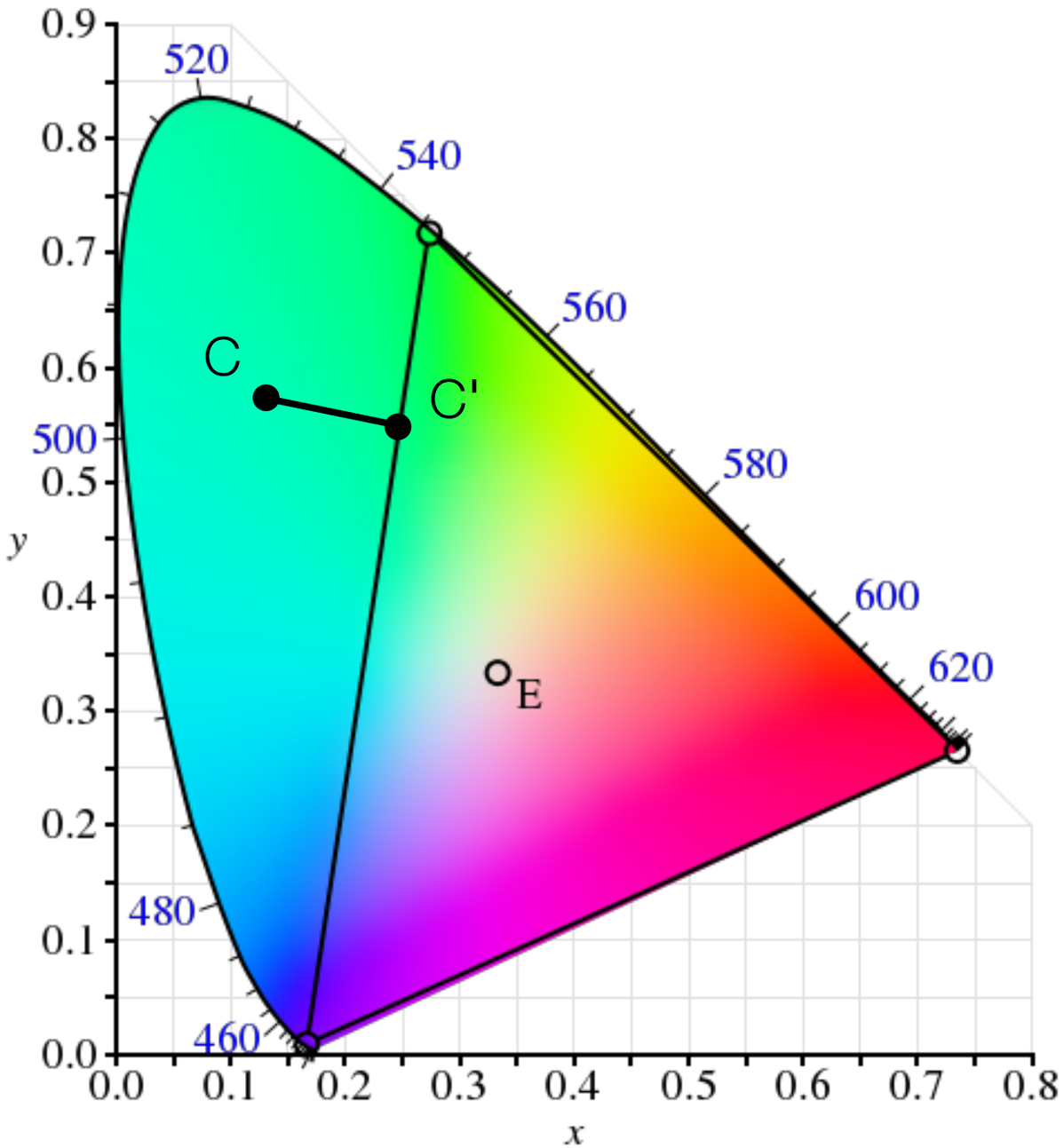


There are four standard rendering intents which describe approaches to gamut mapping.

The definitions are informal.

Implementations vary.

Absolute colormetric

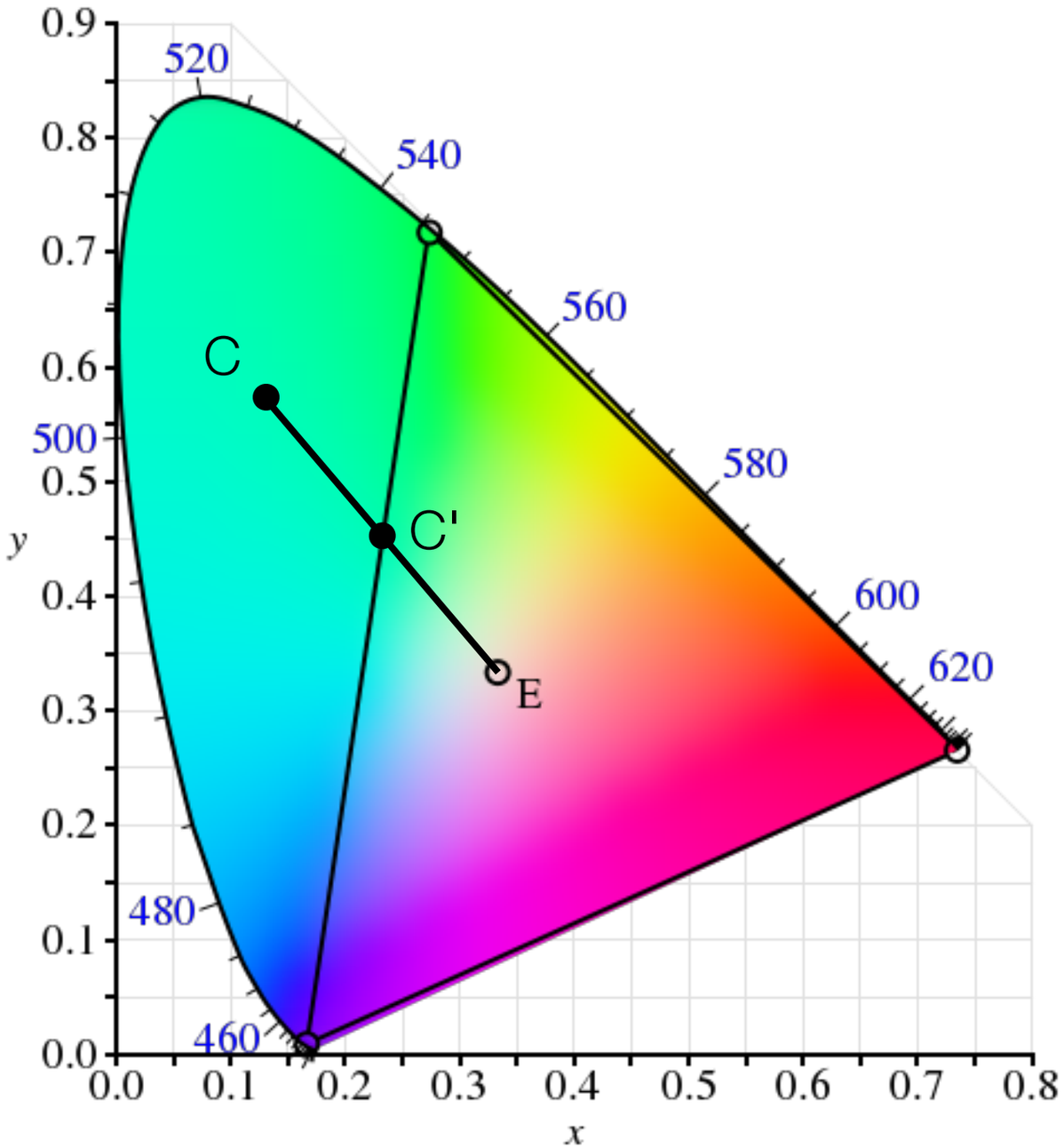


Map C to the nearest point within the gamut.

Distorts hues.

Does not preserve relative saturation.

Relative colormetric

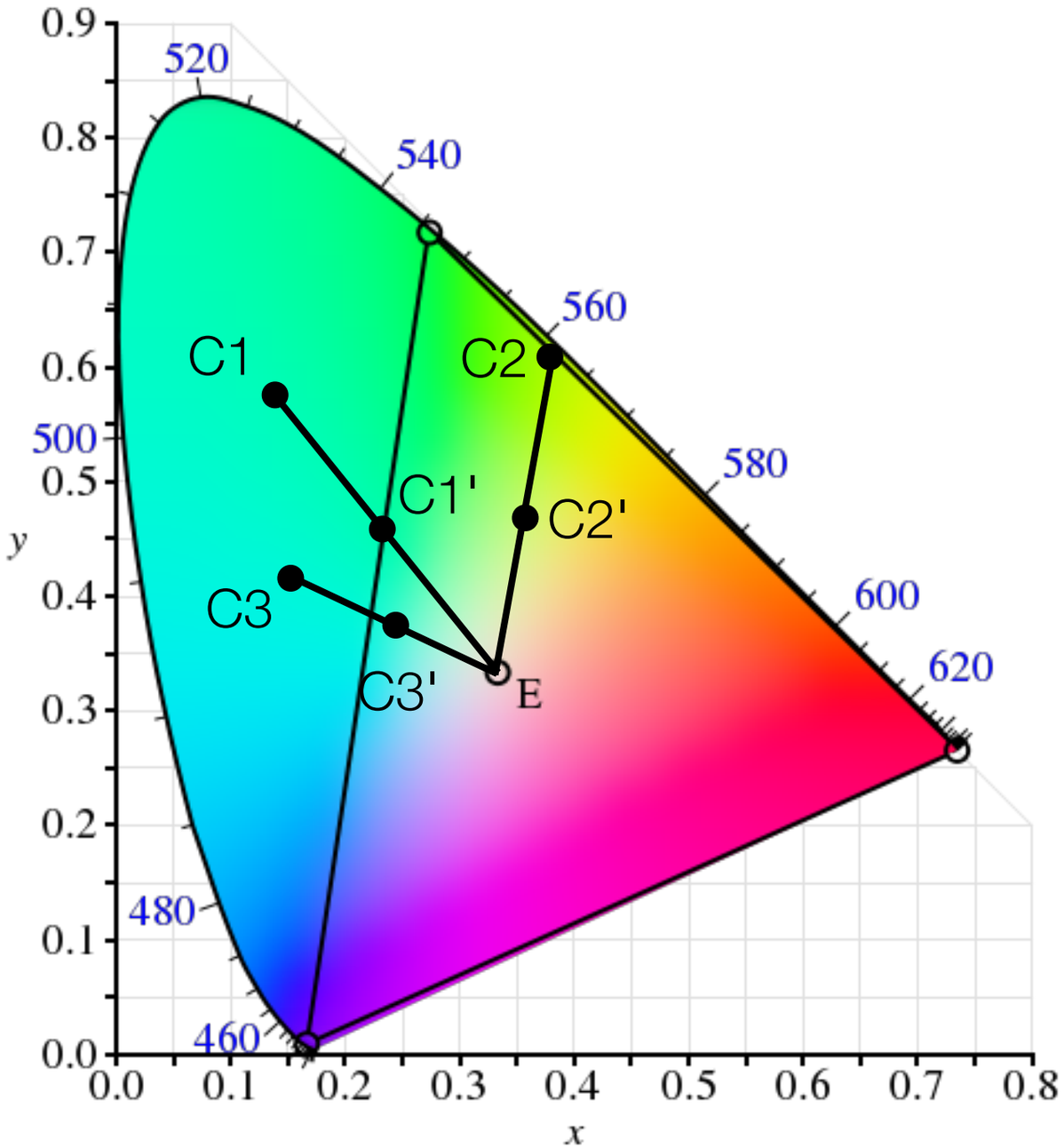


Desaturate C until it lies in the gamut.

Maintains hues more closely.

Does not preserve relative saturation.

Perceptual



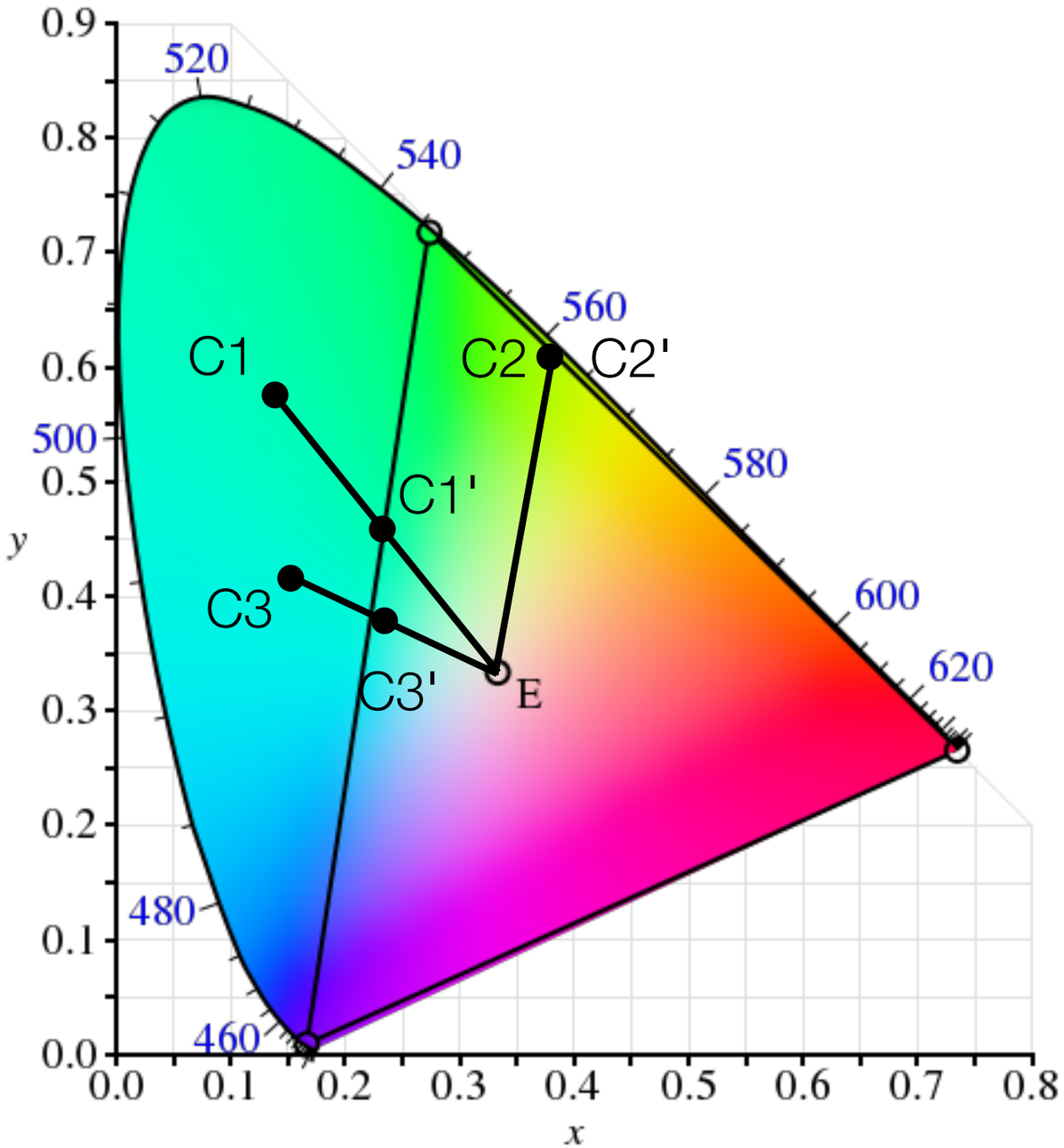
Desaturate all colours until they all lie in the gamut.

Maintains hues.

Preserves relative saturation.

Removes a lot of saturation.

Saturation



Attempt to maintain saturated colours.

Not standardised

Demo

http://graphics.stanford.edu/courses/cs178/applets/gamutmapping.html

Colour space

Standard colour representations:

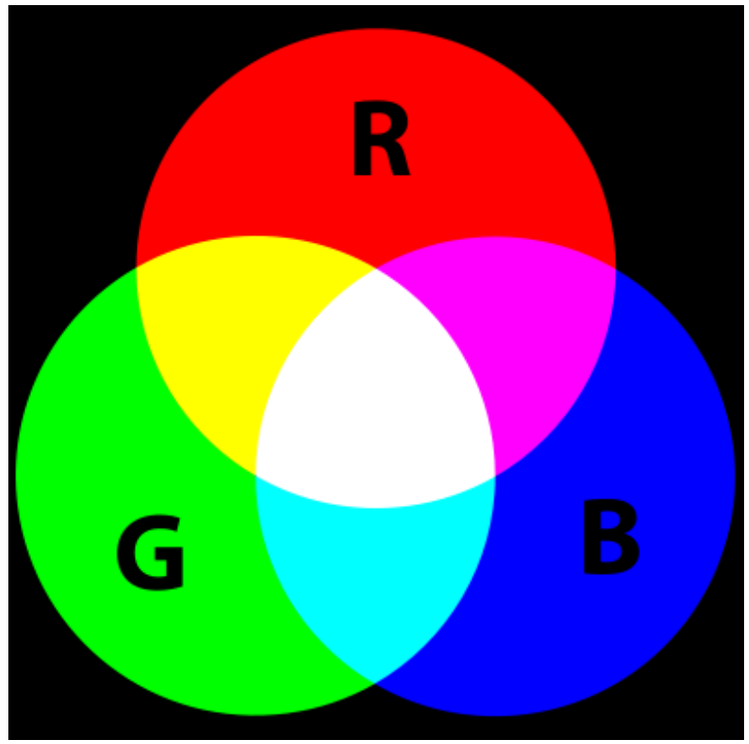
- RGB = Red, Green, Blue
- CMYK = Cyan, Magenta, Yellow, Black
- HSV = Hue, Saturation, Value (Brightness)
- HSL = Hue, Saturation, Lightness

RGB

Colour is expressed as the addition of red, green and blue components.

$$C(r, g, b) = rR + gG + bB$$

This is called additive colour mixing. It is the most common model for computer displays.



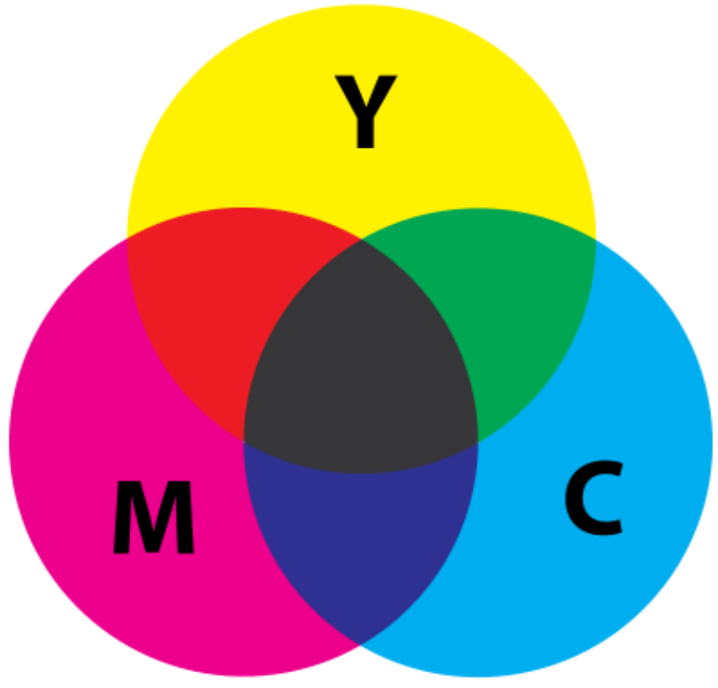
CMY

CMY is a subtractive colour model, typically used in describing printed media.

Cyan, magenta and yellow are the contrasting colours to red, green and blue respectively. I.e.:

$$Cyan = White - Red$$

Cyan pigment/ink absorbs red light.



CMYK

Real coloured inks do not absorb light perfectly, so darker colours are achieved by adding black ink to lower the overall brightness.

The K in CMYK stands for "key" and refers to black ink.

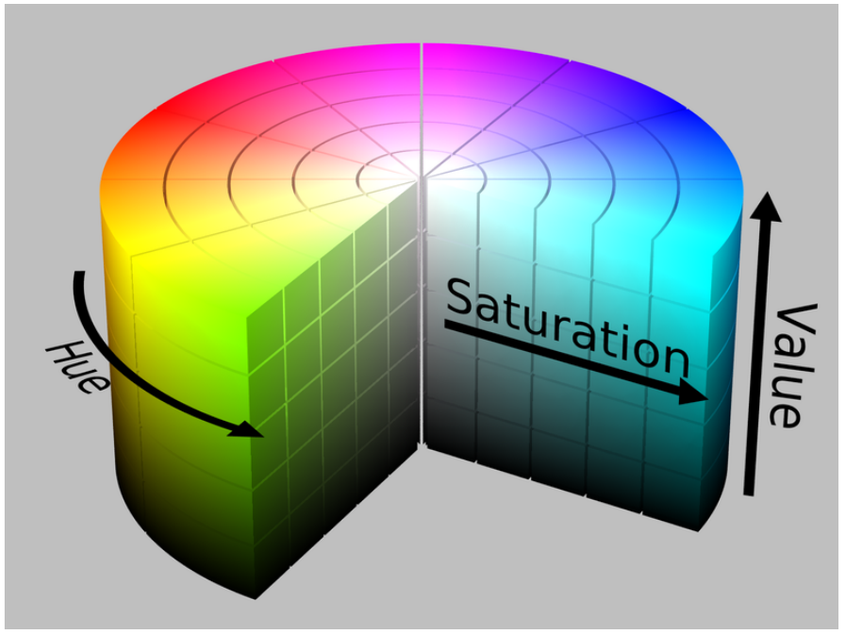
HSV

HSV (aka HSB) is an attempt to describe colours in terms that have more perceptual meaning (but see earlier proviso).

H represents the hue as an angle
from 0° (red) to 360° (red)

S represents the saturation
from 0 (grey) to 1 (full colour)

V represents the value/brightness
form 0 (black) to 1 (bright colour).



HSL

HSL (aka HLS) replaces the brightness parameter with a (perhaps) more intuitive lightness value.

H represents the hue as an angle
from 0° (red) to 360° (red)

S represents the saturation
from 0 (grey) to 1 (full colour)

L represents the
lightness form 0 (black) to 1 (white).