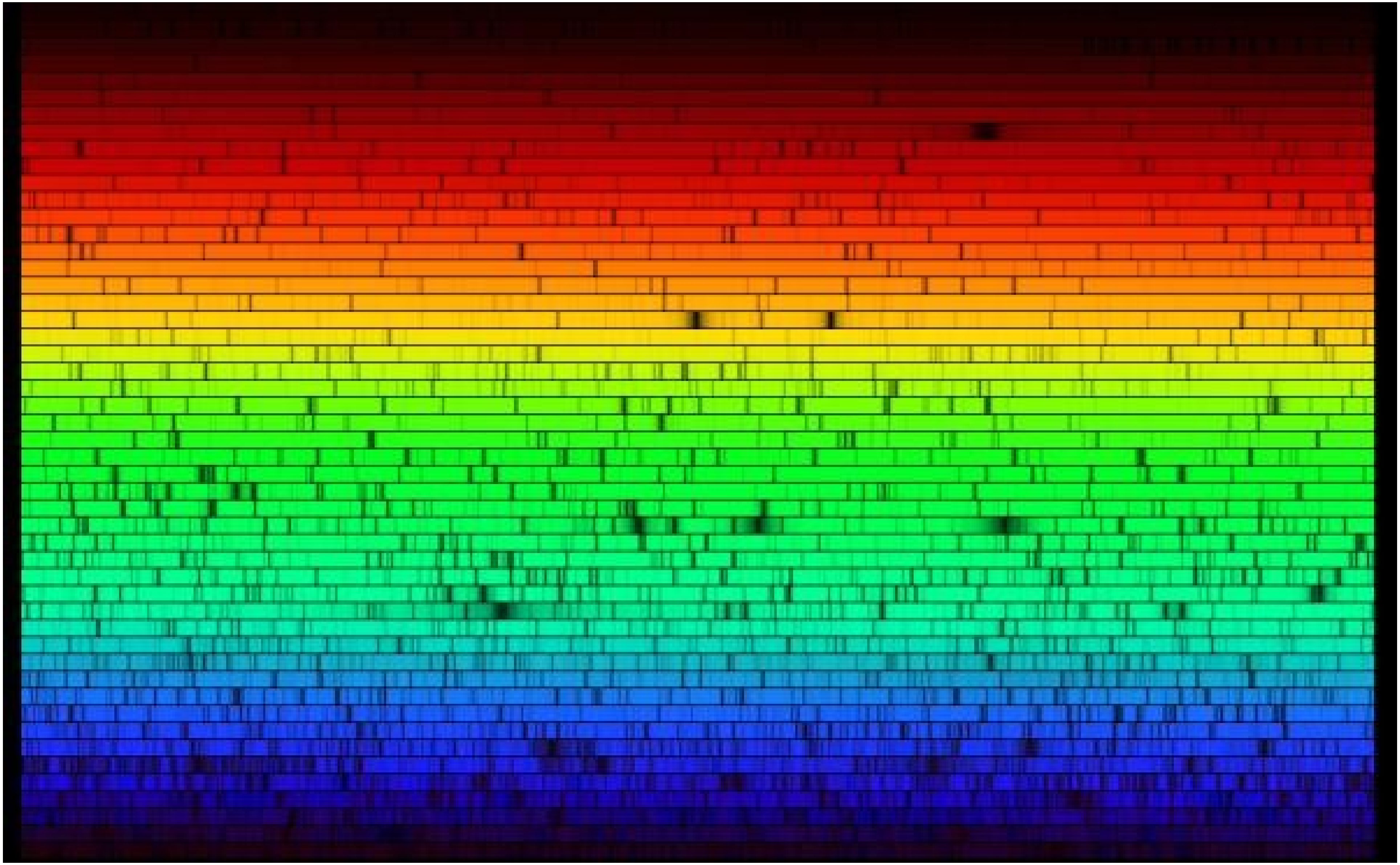
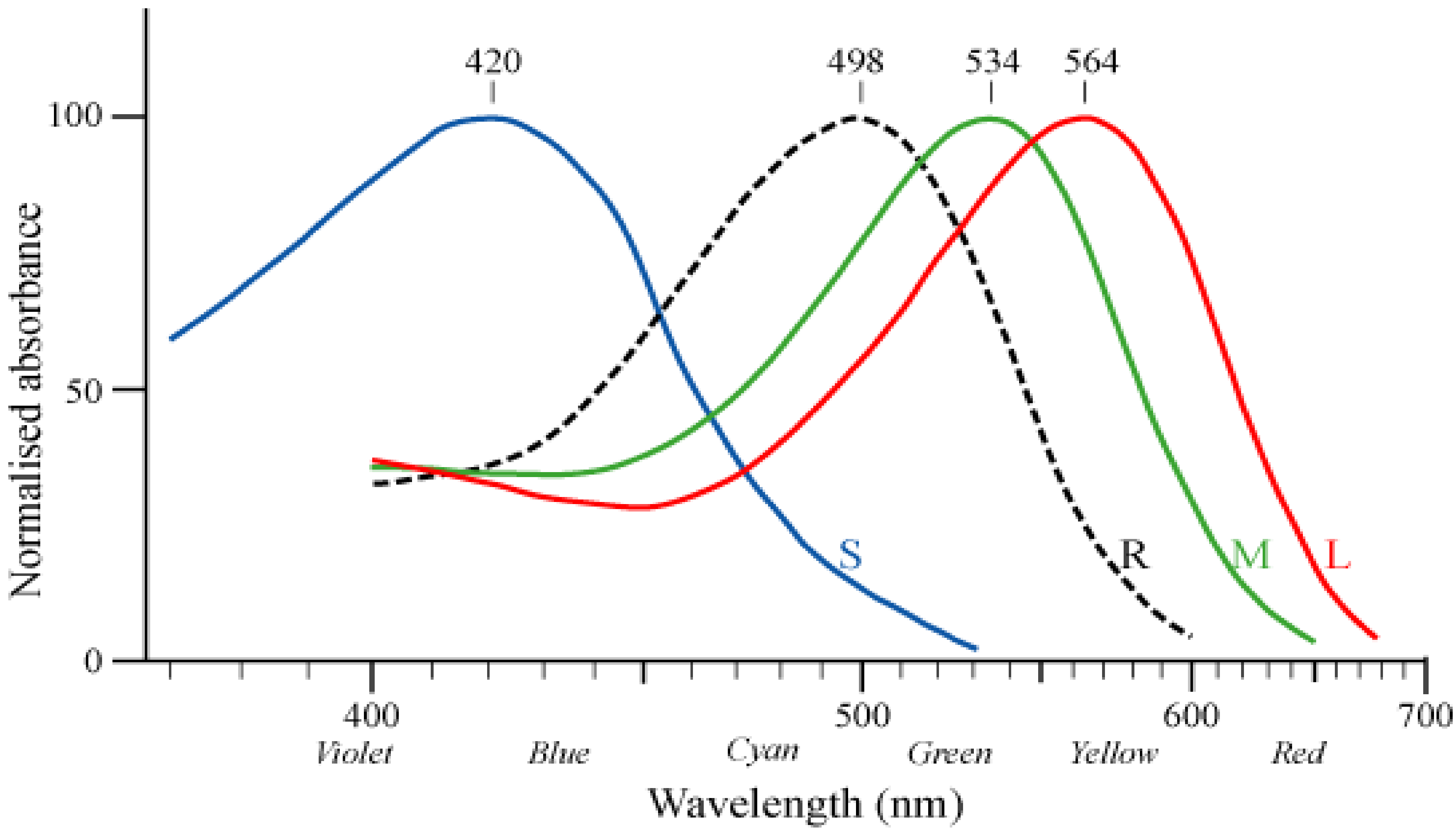
Color and Light

CSCI 4229/5229
Computer Graphics
Summer 2013

Solar Spectrum

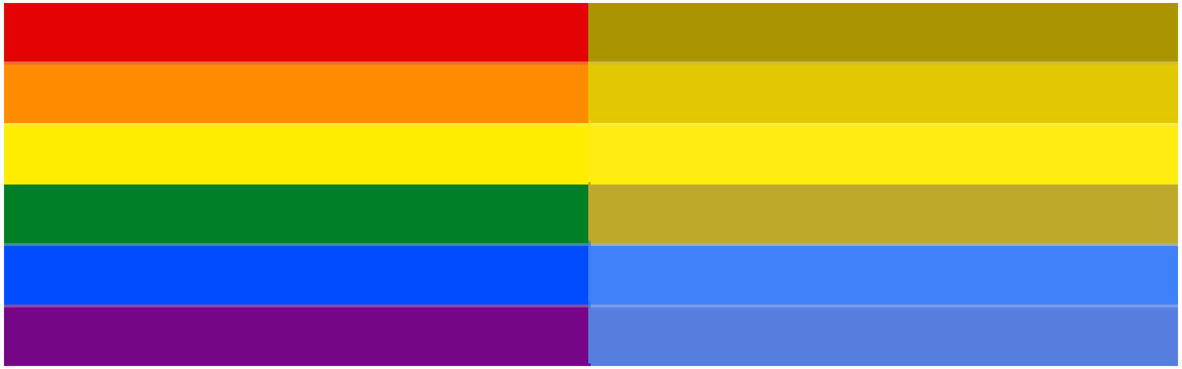


Human Trichromatic Color Perception

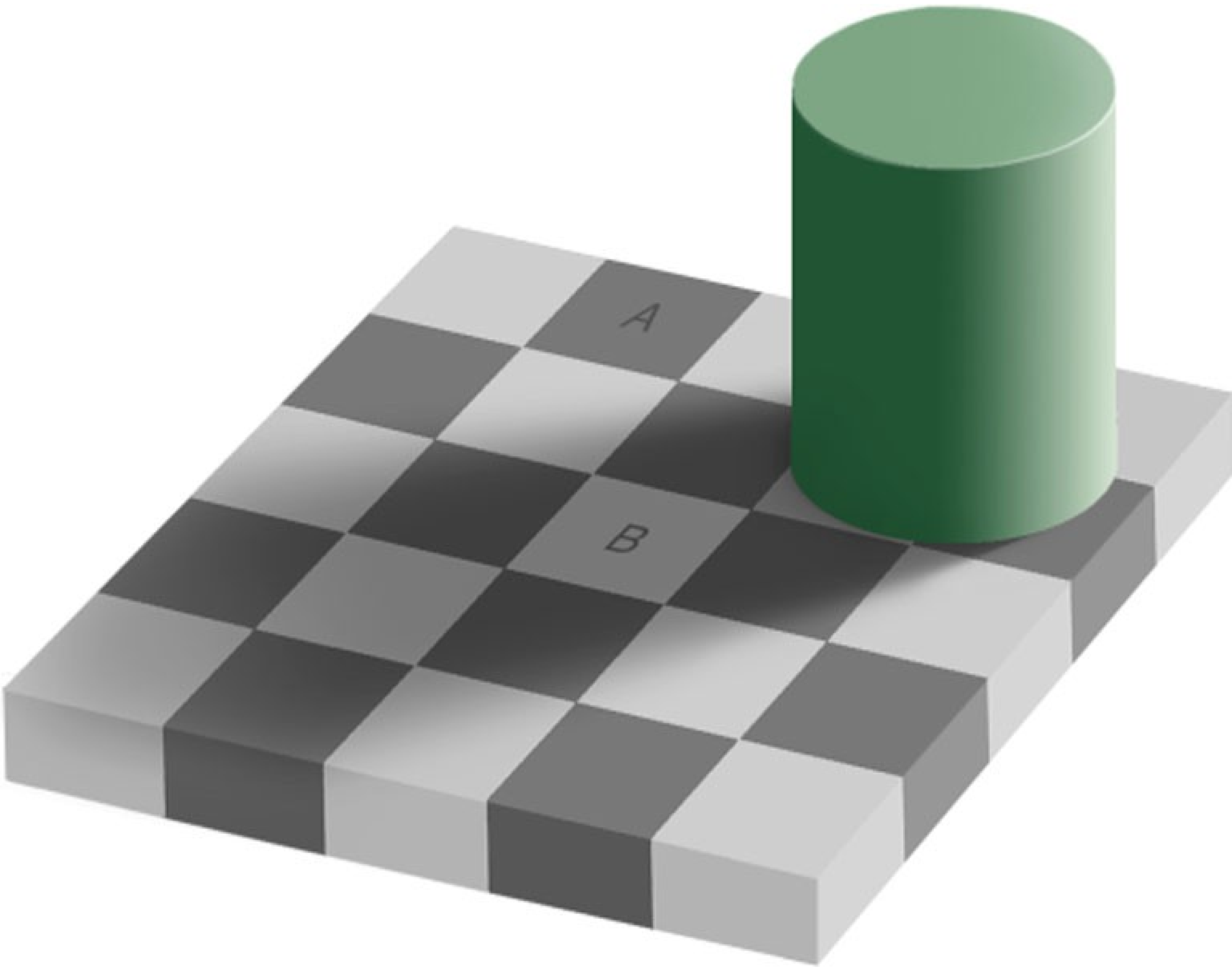


Color Blindness

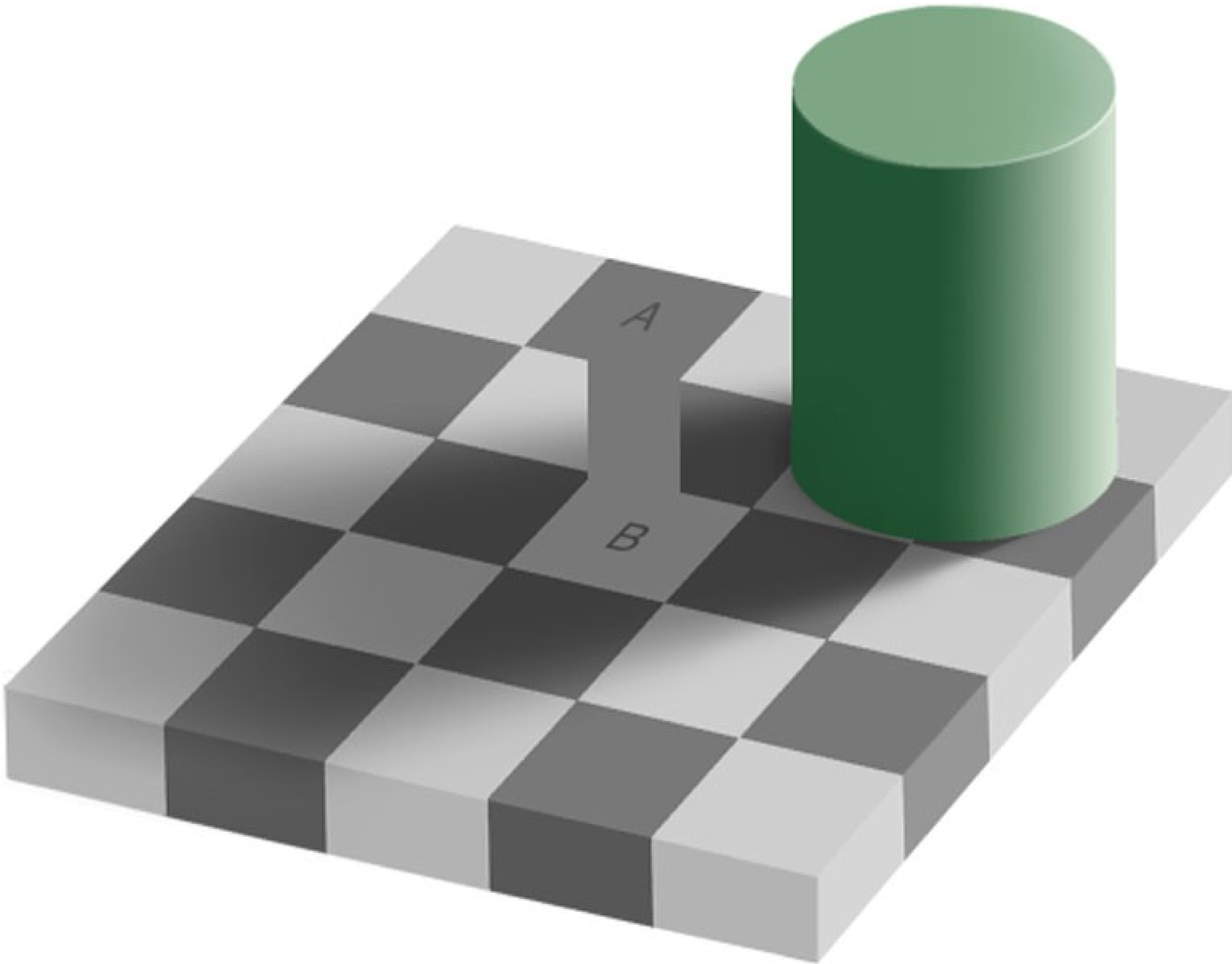
- Present to some degree in 8% of males and about 0.5% of females due to mutation of the X chromosome
- Example of severe red-green color blindness caused by absence of green cone photo receptors



Are A and B the same?



Color perception is relative



Transmission,Absorption&Reflection

- Light source generates radiation with specific energy-frequency spectrum
- Opaque objects absorb some frequencies and reflect others
- Translucent objects absorb some frequencies and transmit others
- Apparent color depends on the spectrum that remains when it reaches the eye

Color Examples

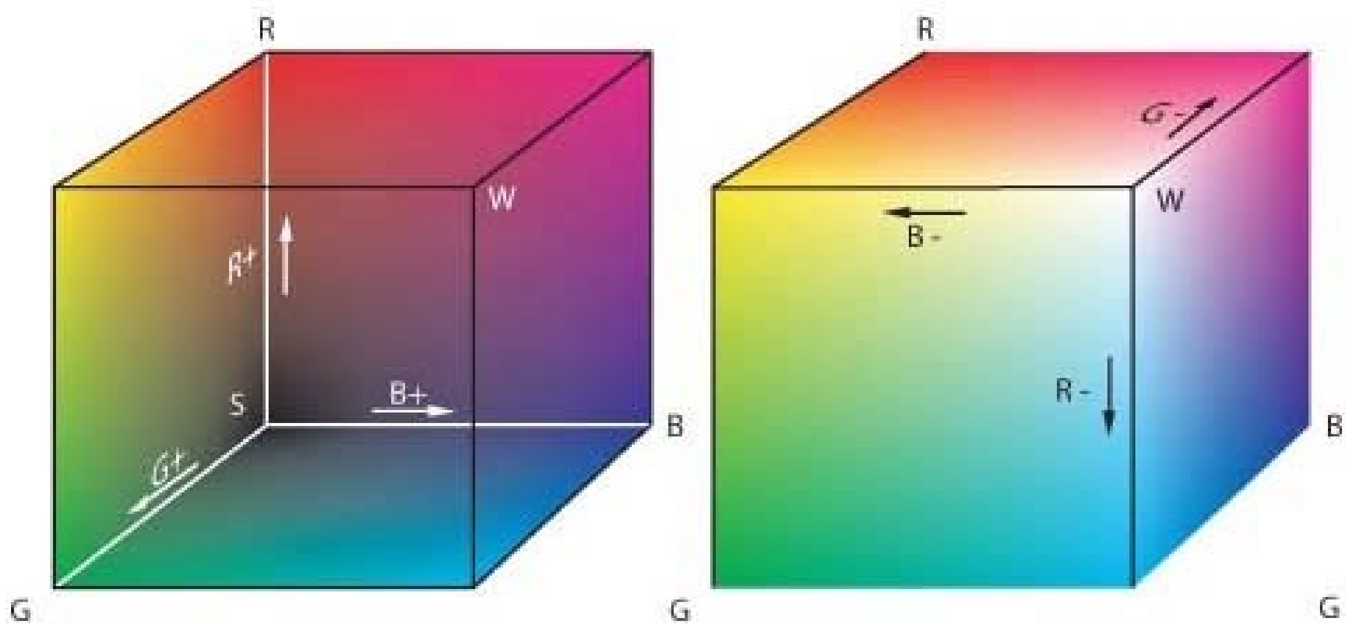
- White source -> red glass -> white paper = red
- White source -> white paper -> red glass = red
- Red source -> white paper = red
- White source -> red glass -> green paper = dark
- Red source -> green glass = dark

RGB Color

- Approximates how humans see
- Additive color
  - Red+Green = Yellow
  - Green+Blue = Cyan
  - Red+Blue = Magenta

Red+Green+Blue = White

No emission = Black

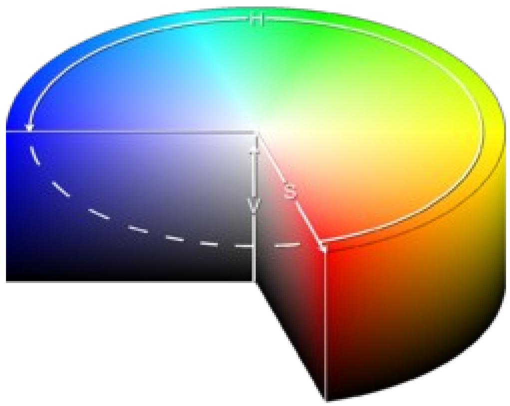


CMY/CMYK Color

- Printing color
- Subtractive Color
  - Yellow+Magenta = Red
  - Yellow+Cyan = Green
  - Cyan+Magenta = Blue
  - Yellow+Cyan+Magenta = Black
  - No ink = White
  - Black helps make darker colors and true black

HSV Color

- Color Progression model
  - Hue (color)
  - Saturation (intensity)
  - Value (brightness)
- Useful in translating values to range of colors



Color in the Real World

- Sunlight is essentially white
  - Incandescent light is yellowish
  - Fluorescent light is mostly blue-green
- Reflected light depends on the surroundings
  - Wall, ceiling and floor color
  - Large objects
  - Filtered light
- Light is often bounced off multiple objects before it reaches the eye

Local vs. Global Lighting

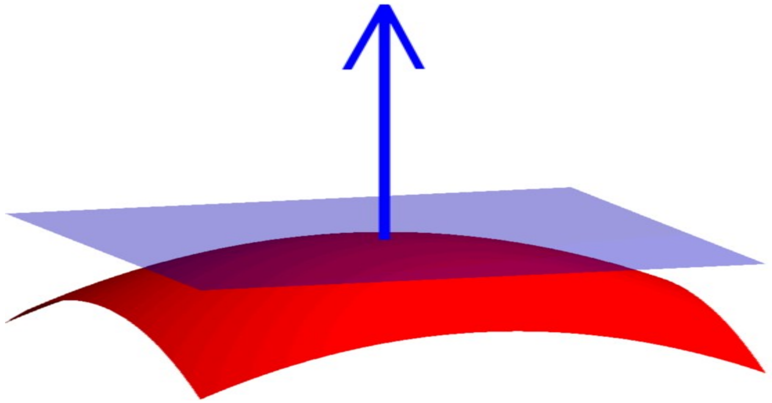
- Global lighting
  - Traces how light bounces off successive objects
  - Recursive Ray tracing and Radiosity
  - Not currently practical for real time graphics
- Local lighting
  - Separates light sources into direct and ambient
  - Calculates intensity based only on vectors
  - Many possible simplifications
  - Requires special action to generate shadows

Color and Materials

- Mirrors reflect (almost) all light
  - Highly directional
- Metals, glazed ceramics, calm water, ...
  - Mostly directional, some diffuse
- Plastics, unglazed ceramics, turbulent water, ...
  - Some directional, mostly diffuse
- Natural materials (leaves, leather, skin, ...)
  - Predominantly diffuse

Surface Normals

- How light interacts with a surface depends on the angle between the light rays and the surface
- The vector perpendicular to the surface is called the ***surface normal*** or just ***normal*** vector
- For a flat surface the normal is the same for all points on the surface
- For a curved surface the normal is potentially different at every point



Determining Surface Normals

- By inspection
  - Cube (parallel to axes)
  - Sphere (radially out)
  - Cone (singularity at top, radially out and up)
  - Torus (radially out from central axis)
- Cross product of tangental vectors
  - Polygons (difference between vertexes)
  - Analytical derivatives
- Use normals of actual surface, not polygons used to approximate the surface (see Gouraud shading)

Lighting for Real Time Graphics

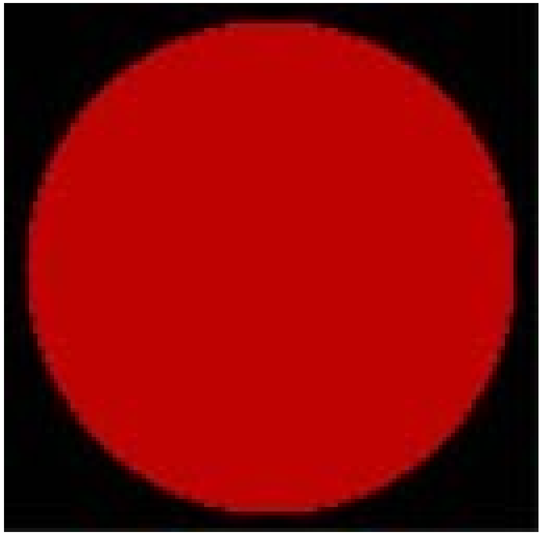
- Add different “types” of light
  - Ambient: Scattered light from diffuse sources
  - Diffuse: Scattered light from point sources
  - Specular: Directional light from point sources
  - Emission: Light from object itself
- Local calculations: Light interacts with objects
- Lighting calculations by component (RGB)
- Lighting done by vertex
  - Use must specify an appropriate normal

Emission Light

- Object radiates light in all directions
- Color and intensity independent of incident light
- Typically used to represent internally lit objects
- Intensity M_E (which is a property of the material)
- *If calculations are LOCAL the light from internally lit objects does not light up the scene*
    - *This is what OpenGL does*

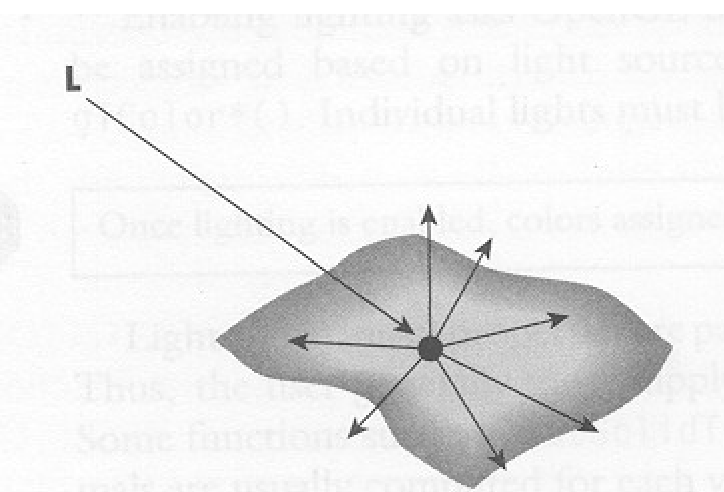
Ambient Light

- Light comes from all directions and is reflected in all directions
- The color and intensity of ambient light represents the net result of multiple reflections
  - Color of walls illuminated by white light
  - Color of canopy in forest
- Intensity MC_A



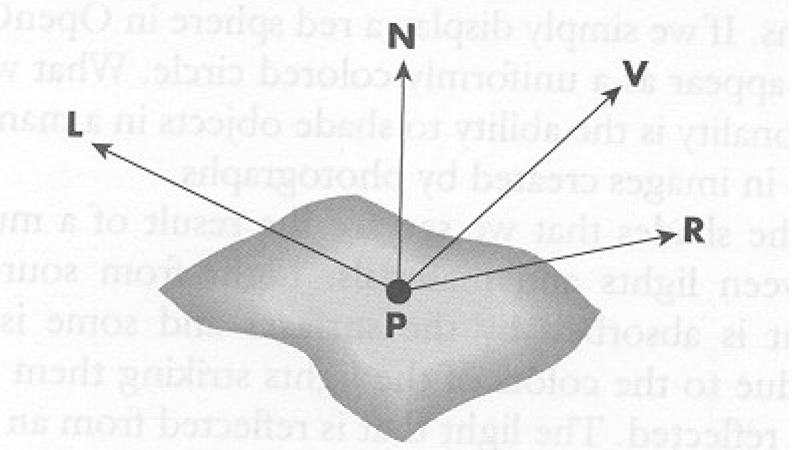
Diffuse Reflections

- Lambertian Reflection
- Diffuse light scatters in all directions
- Intensity depends on cosine of the angle of incidence
- Intensity $(N \bullet L) M C_D$



Specular Reflection

- Light is reflected in a preferred direction
- Responsible for bright spot on spheres
- Direction of reflection is $R = 2(L \bullet N)N - L$
- Intensity does not change along R
  - Gets dimmer away from R



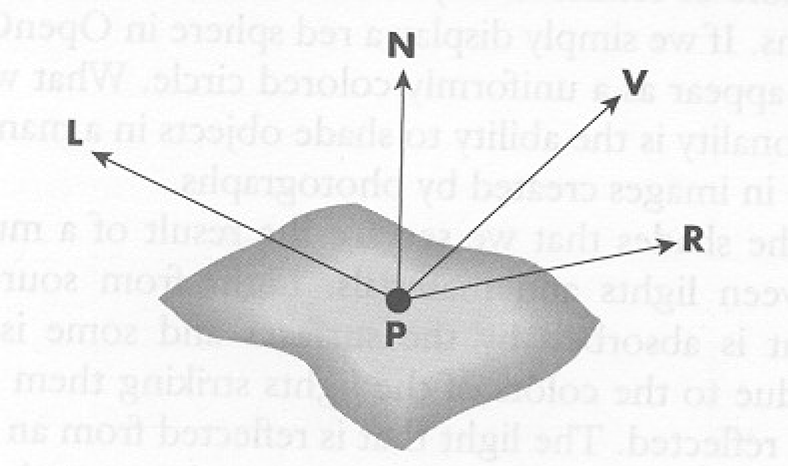
Phong Reflection Model

Bui Tuong Phong, University of Utah, 1973

- L light source
- N normal vector for surface
- R reflected light

  $R = 2(L \bullet N)N - L$

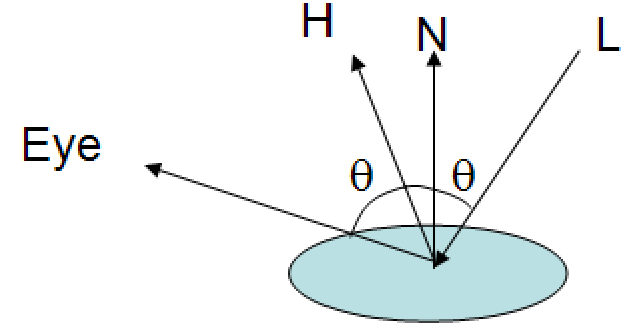
- V viewer (eye)
- Intensity $(V \bullet R)^S MC$
  - S shininess
  - M material reflection coefficient
  - C color if light source
- Calculated independently for R,G,B



Blinn-Phong Reflection Model

Jim Blinn, University of Utah, 1978

- Also called modified Phong or Fast Phong
- Half angle $H = L+V$ (renormalize)
- Simpler and faster
  - Often V = (0,0,1)
  - L constant if far
- Intensity $(N \bullet H)^{S} MC$



Gouraud Shading

Henri Gouraud, University of Utah, 1971

- Calculate lighting effects only at vertices
  - True Gouraud shading calculates vertex normal as the average of the normals of adjoining polygons
- Interpolate lighting effects (colors) across the surface of the polygon
- Potentially significant computational savings over calculating lighting effects for each pixel

Local Light Calculations

- Light is additive
  - Color intensity is the sum of all light sources and types (ambient, specular, ...)
  - Colors are added by component (R,G,B separately)
  - Color intensity varies with the cosine of angles
  - Color hue changes as relative component intensities change
- Light behaves algebraically
  - Intensity varies essentially linear
  - Shadows can be made by subtracting light

OpenGL Light Types

- Global Ambient
  - Ambient light not from any specific light source
- Ambient
  - Light from all directions associated with source
- Diffuse
  - Light reflecting in all directions
- Specular
  - Light reflecting in preferred direction
- Emission
  - Light emanating from each object

Combined OpenGL Lighting

- Color = $M_E + M_A C_A + (N \cdot L) M_D C_D + (N \cdot H)^S M_S C_S$
- Calculated for each light, vertex, RGB
- Requires normalized (0-1) values
- User must specify
  - Light color and position
  - Object color and normal

Physics of Non-directional Light

- Point source light
  - Radiates in all directions
  - Intensity decays inversely proportional to r^2
  - Absorption could attenuate light faster
- Diffuse surface reflection
  - Radiates in normal hemisphere
  - Reflected from rough (matte) surface
  - Also absorbed and re-emitted light

Hints on Using Lights

- Stick to a single light
- Use white lights (R=G=B=α=1)
- Specify one ambient light (global or primary)
  - Intensity should be in the 0.1 to 0.3 range.
- Diffuse (soft) light should always be present, intensity 1.
- Specular (hard) light creates highlights, sparkles, etc.

Hints on Using Materials

- Changing color on materials is simpler than changing the color of light
- Use the same color for ambient and diffuse
- Materials are typically one of
  - matte, high diffuse color and low specular
  - plastic, with high diffuse color and white specular
  - metal, with low diffuse and high specular color
- Color is determined by specular for metals, diffuse for other materials

Hints on Surface Normals

- Surface normals are required for lighting
  - glu and glut objects calculate these
- Normals are perpendicular to polygons
- Normals are transformed with verteces
- Normals must be unit length (watch scaling)
  - glEnable(GL_NORMALIZE) enforces this

OpenGL Lighting Controls

- Enable Lighting
- Light Sources
  - glLight*
- Material types
  - glMaterial*
- Normals
  - glNormal*