

## Colour Reproduction

Key point--color reproduction is based on “metamerism”

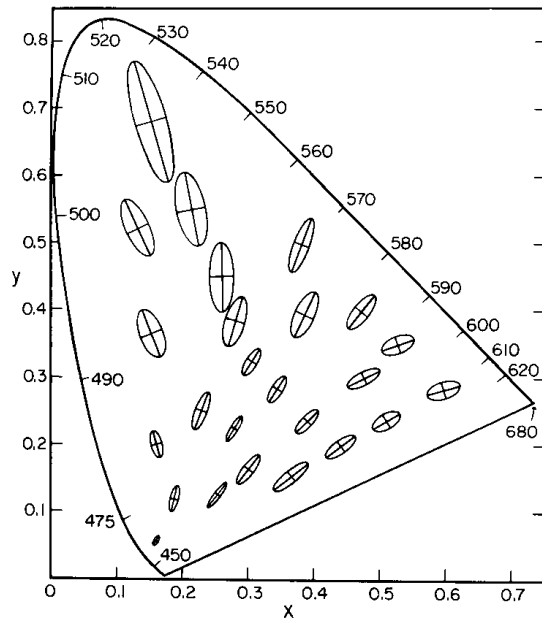
Metameric match--colors which match, despite different spectra.

Duplicating spectra would work, but for practical reasons, we duplicate the match.

For reflective surfaces, e.g prints, this means that the match can change if the illumination changes.

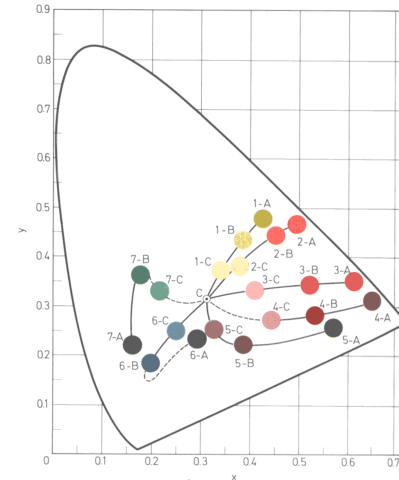
## The quest for uniform colour spaces

- Definition of uniform: equal (small!) steps give the same perceived color changes.
- XYZ is not uniform!
- Uniformity only applies to small differences. There is no theory for numerically deciding if yellow is perceptually closer to green or red.



MacAdam Ellipses  
(scaled by a factor  
of 10) on CIE x, y

## Mixing pigments in CIE



Color matching is linear, but combining pigments is not necessarily linear like mixing light .

## Shading values for colored surfaces

- Simplest:
  - Use appropriate shading model in 3 channels, instead of one
  - Implies red albedo, green albedo, blue albedo, etc.
  - Works because the shading model is independent of wavelength.
  - Can lead to somewhat inaccurate colour reproduction in some cases - particularly coloured light on coloured surfaces
- Better
  - Use appropriate shading model at many different wavelength samples - 7 is usually enough
  - Estimate receptor response in eye using sum over wavelength
  - Set up pixel value to generate that receptor response

## Monitor Gamma

A typical image encoding is **NOT** linear. Often a gamma correction is included. This leads to no end of confusion.

A “gamma” corrected image is ready to drive a CRT monitor, and has advantages that quantization (8 bits) errors are *roughly* uniformly distributed--that fact that this works is a convenient accident.

## Monitor Gamma

Due to the physics involved, CRT monitor brightness is proportional to  $\text{voltage}^{(2.5)}$

This is further hacked to give the “standard” gamma of 2.2

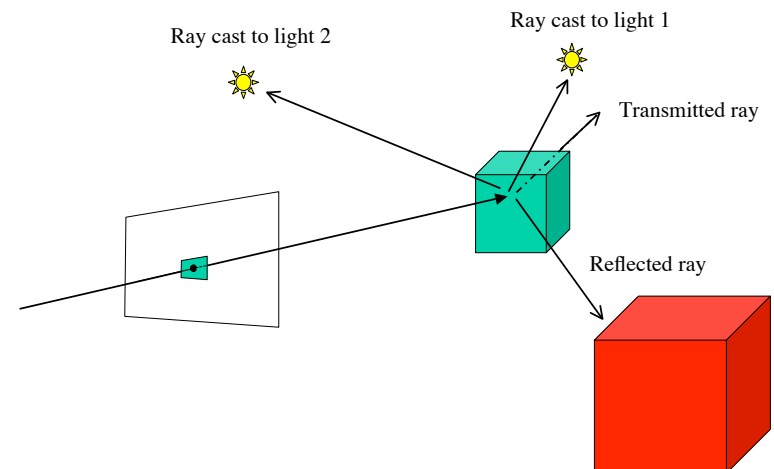
So, if an image looks good on a CRT, it is likely to be non-linear by  $\text{pow}(1/2.2)$

LCD--more linear, but then hardware/software can be hacked to be like CRT

Confusing? Yes!

## Recursive ray tracing

H&B, page 597



## Recursive ray tracing rendering algorithm

- Cast ray from pinhole (projection center) through pixel, determine nearest intersection
- Compute components by casting rays
  - to sources = shadow ray (diffuse and for specular lobe)
  - along reflected direction = reflected ray
  - along transmitted dir = refracted ray
- Determine each component and add them up with contribution from ambient illumination.
- To determine some of the components, the ray tracer must be called **recursively**.

## Recursive ray tracing rendering (cont)

- Recursion needs to stop at some point!
- Contributions die down after multiple bounces---there is no such thing as a perfect reflector---so we either set mirror reflections to be less than 100% (even if the user asks for 100%), or simply include an attenuation factor for each new ray.

- Can also model absorption due to light traveling in medium
  - Usually ignored in air, but depends on the application
  - Translucent absorption is exponential in depth

$$I = I_0 e^{-\alpha d}$$

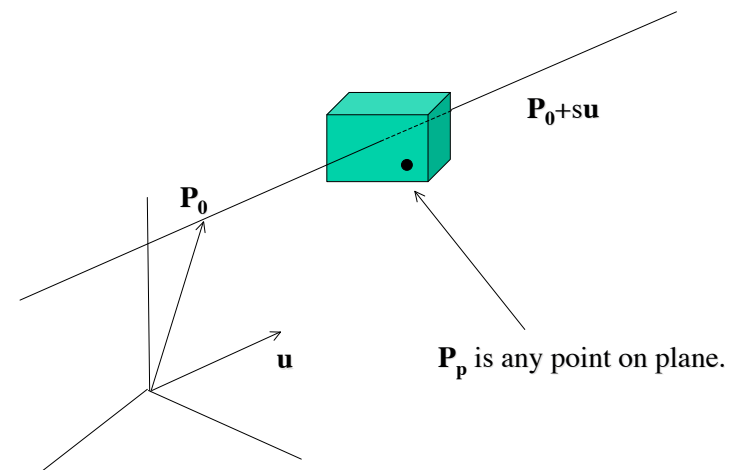
- Recursion is stopped when contributions are too small
  - need to track the cumulative effect
  - common to also limit the depth explicitly

## Mechanics

- Primary issue is intersection computations.
  - E.g. sphere, triangle.
- Polygon (should feel familiar!)
- Find point on plane of polygon and then determine if it is inside
  - One way is to make an argument with angles
  - Another way---thinking of the polygon as a surface of a polyhedra---is to check if the point is on the inside side of each of the other planes of the polyhedra.
- Sphere, relatively simple algebra.

## Poly details

May be helpful for A6.



## Poly details

May be helpful for A6.

To find the intersection of the ray and the plane, solve:

$$(\mathbf{P}_0 + s\mathbf{u} - \mathbf{P}_p) \cdot \mathbf{n} = 0$$

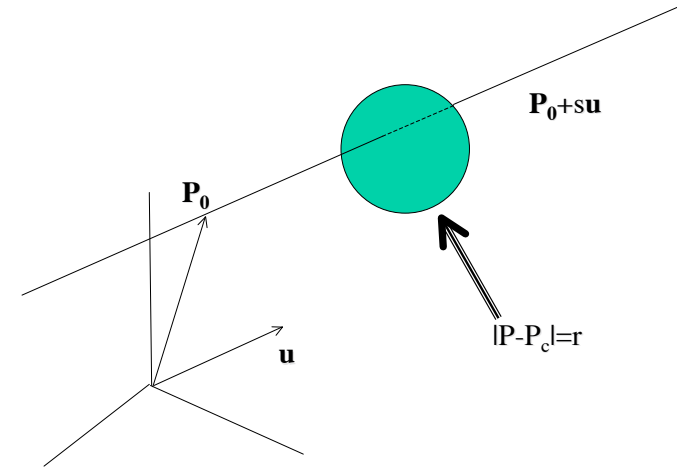
Once you have the point of intersection,  $\mathbf{P}_i$ , test that it is inside by testing against all other faces.

$$(\mathbf{P}_i - \mathbf{P}_p) \cdot \mathbf{n} < 0$$

Note that  $\mathbf{n}$  and  $\mathbf{P}_p$  are now from those *other* faces.

## Sphere details (H&B, 602)

May be helpful for grad version of A6.



## Sphere details (H&B, 602)

May be helpful for grad version of A6.

$$|\mathbf{P}_0 + s\mathbf{u} - \mathbf{P}_c| = r$$

$$|\Delta\mathbf{P} + s\mathbf{u}| = r$$

$$(\Delta\mathbf{P} + s\mathbf{u}) \cdot (\Delta\mathbf{P} + s\mathbf{u}) = r^2$$

$$\Delta\mathbf{P} \cdot \Delta\mathbf{P} - r^2 + 2s\Delta\mathbf{P} \cdot \mathbf{u} + s^2\mathbf{u} \cdot \mathbf{u} = 0$$

The last expression is easily solved using the quadratic equation. If the discriminant is negative (complex solutions), then the ray does not intersect the sphere.

## Sphere details (H&B, 602)

May be helpful for grad version of A6.

Recall that if:  $as^2 + bs + c = 0$

The “discriminant” is:  $b^2 - 4ac$

The solution is:  $s = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

Note that in the book,  $\mathbf{u}$  is a unit vector, so  $\mathbf{u} \cdot \mathbf{u} = 1$ , thus  $a=1$ , and  $b$  has a factor of 2 that is removed by dividing by  $2a=2$ , to get equation 10-71.

## Refraction Details

Index of refraction,  $n$ , is the ratio of speed of light in a vacuum, to speed of light in medium.

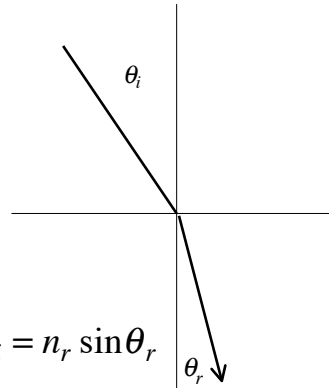
Typical values:

air:	1.00 (nearly)
water:	1.33
glass:	1.45-1.6
diamond:	2.2

$$n_i \sin \theta_i = n_r \sin \theta_r$$



The indices of refraction for the two media, and the incident angle,  $\theta_i$ , yield the refracted angle  $\theta_r$ . (Also need planarity).



## Recursive ray tracing

H&B, page 597

