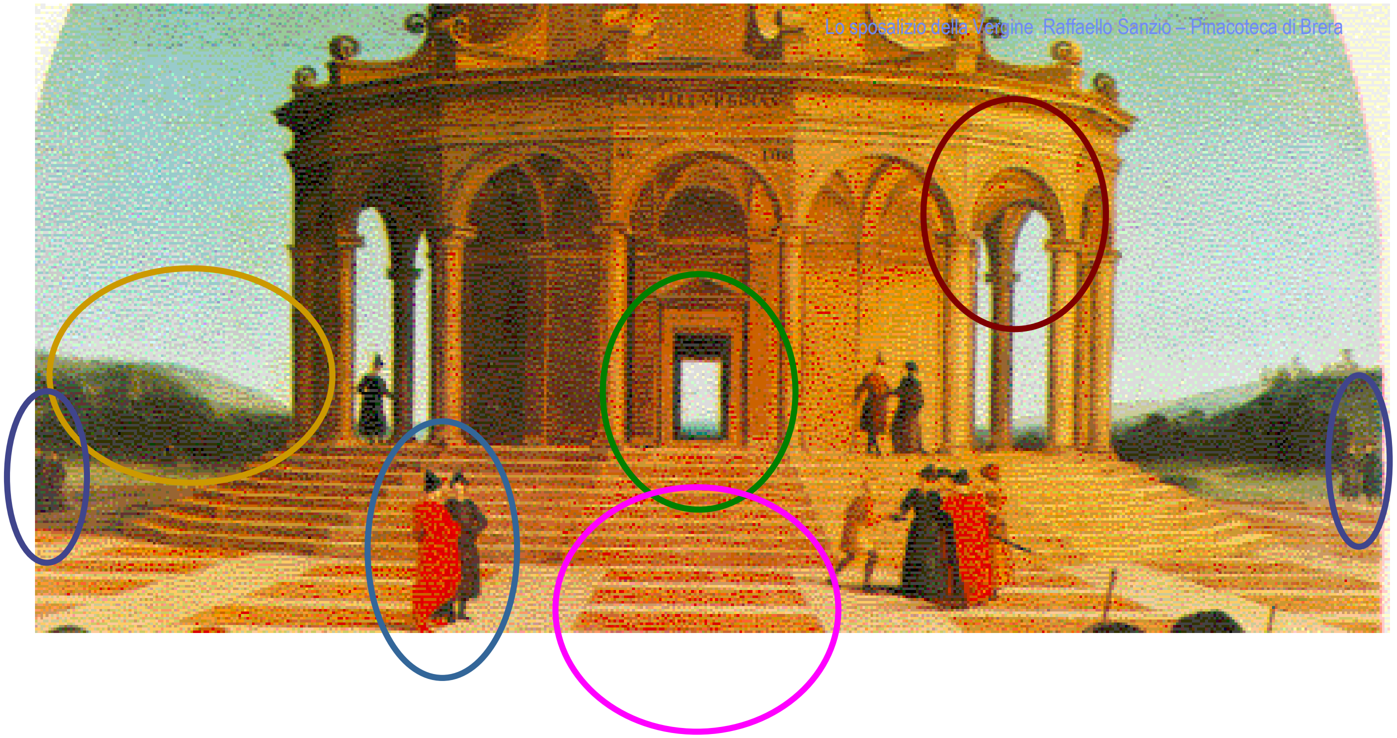
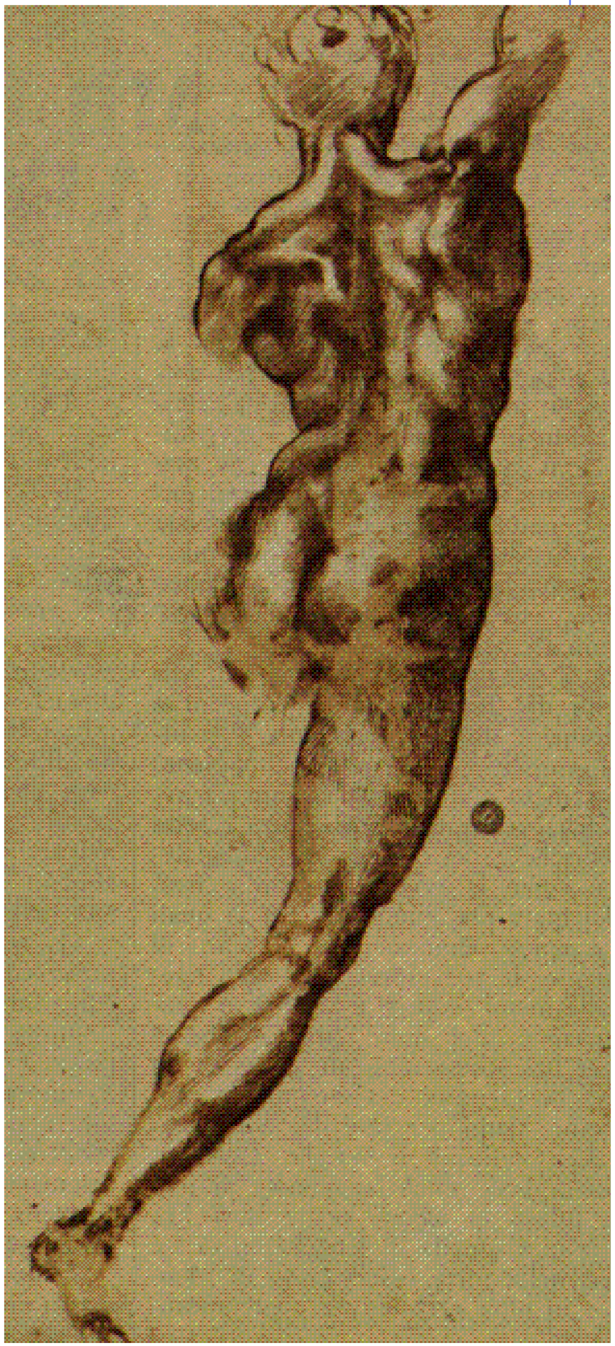
The photometric track

Raffaello Sanzio, Lo sposalizio della Vergine,
1504, Milano, Pinacoteca di Brera


Lo sposalizio della Vergine, Raffaello Sanzio – Pinacoteca di Brera



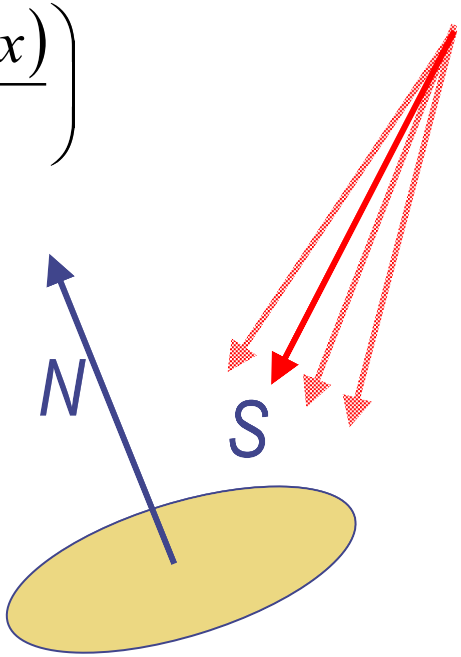
Michelangelo, 1528

Standard nearby point source model

- N is the surface normal, ρ is diffuse albedo, S is source vector whose length is the intensity term

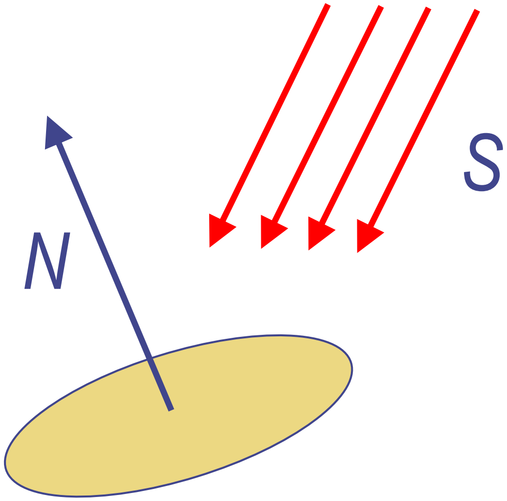
Nearby point source model

$$\rho_d(x)\left(\frac{N(x)\cdot S(x)}{r(x)^2}\right)$$



Distant point source model

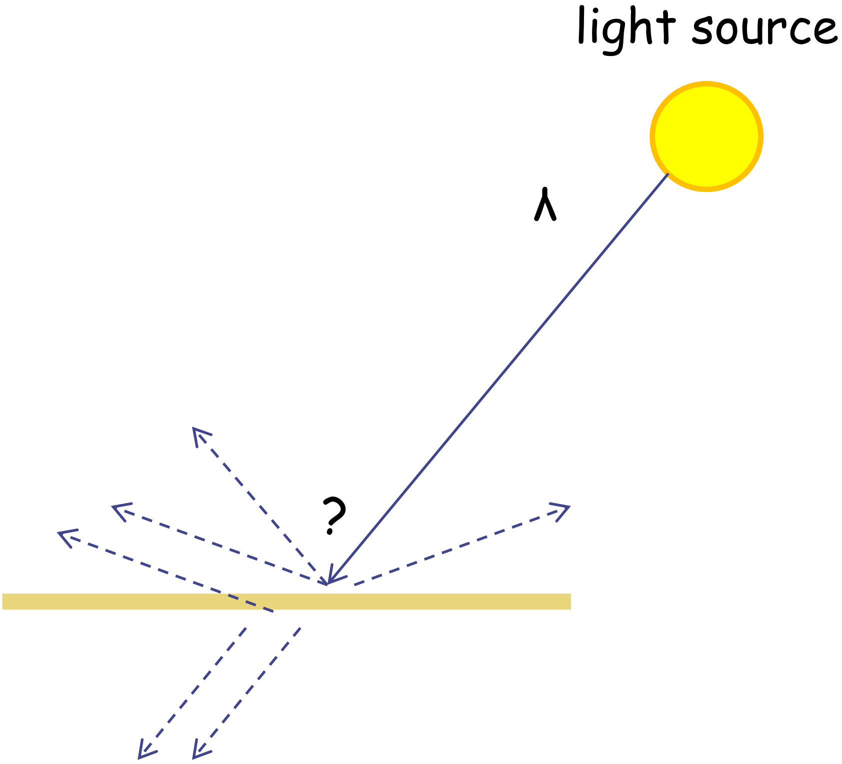
$$\rho_d(x)(N(x)\cdot S_d(x))$$



Assume that all points in the model are close to each other with respect to the distance to the source. Then the source vector doesn't vary much, and the distance doesn't vary much either, the computation are simplified.

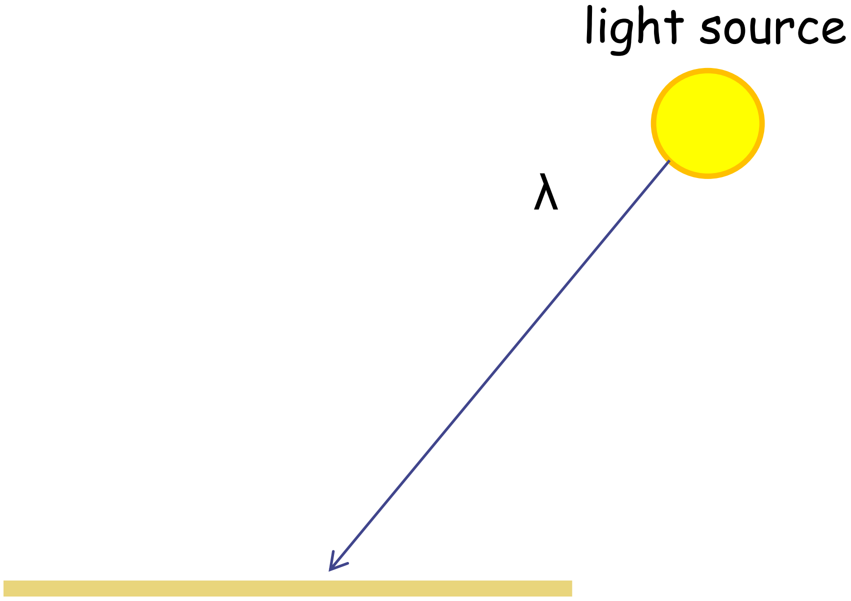
A photon's life choices

- Absorption
- Diffusion
- Reflection
- Transparency
- Refraction
- Fluorescence
- Subsurface scattering
- Phosphorescence
- Inter-reflection



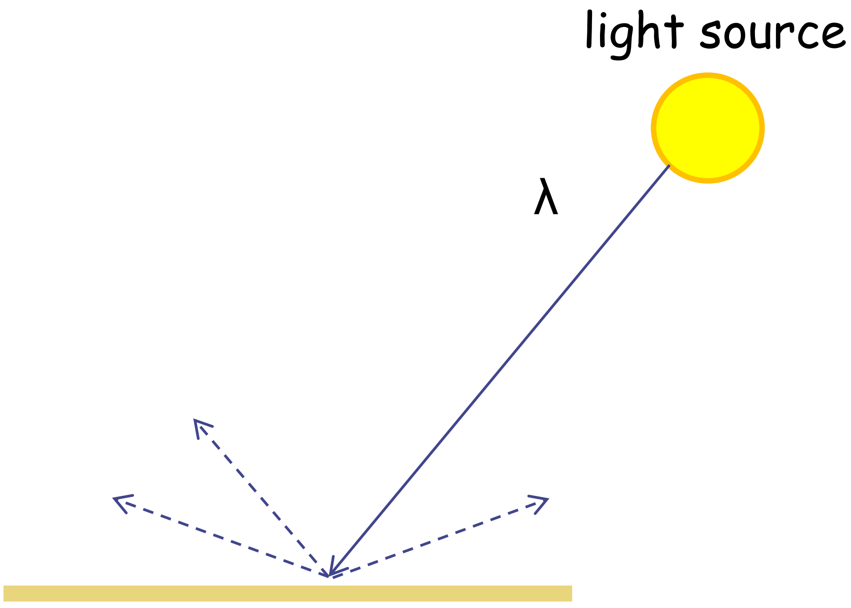
A photon's life choices

- **Absorption**
- Diffusion
- Reflection
- Transparency
- Refraction
- Fluorescence
- Scattering
- Phosphorescence
- Inter-reflection



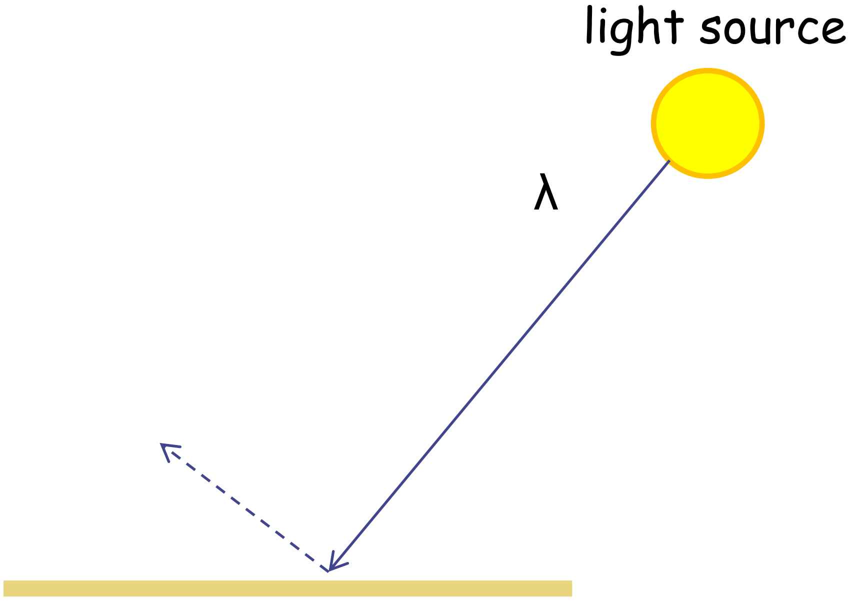
A photon's life choices

- Absorption
- **Diffuse Reflection**
- Reflection
- Transparency
- Refraction
- Fluorescence
- Scattering
- Phosphorescence
- Interreflection



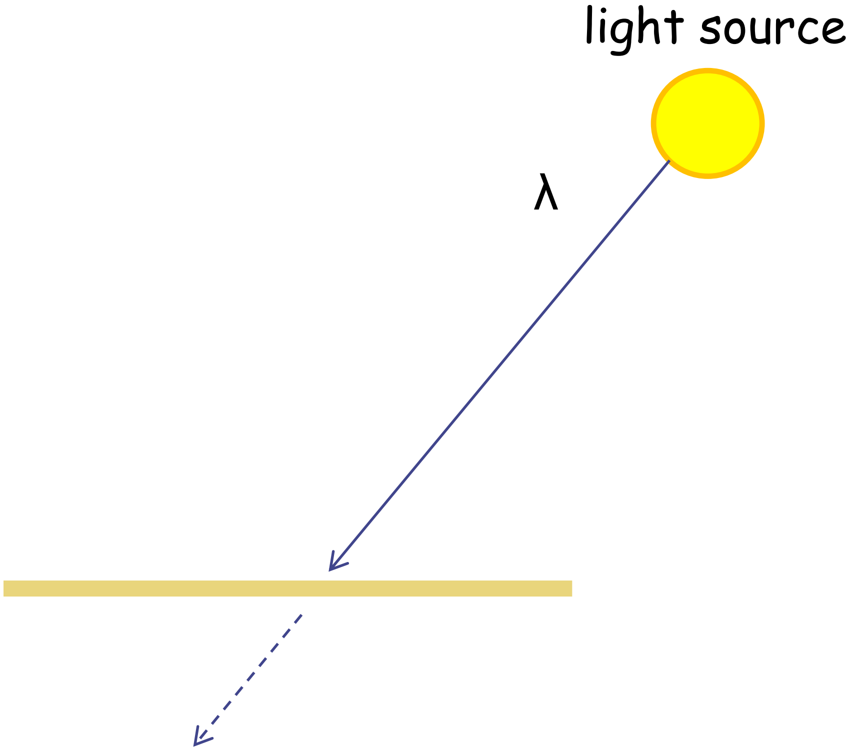
A photon's life choices

- Absorption
- Diffusion
- **Specular Reflection**
- Transparency
- Refraction
- Fluorescence
- Scattering
- Phosphorescence
- Inter-reflection



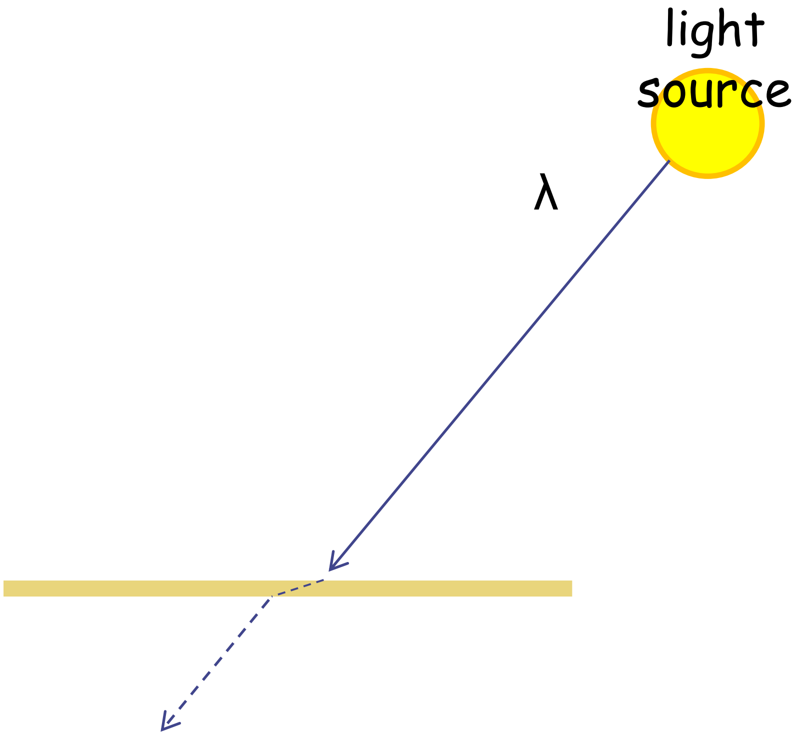
A photon's life choices

- Absorption
- Diffusion
- Reflection
- **Transparency**
- Refraction
- Fluorescence
- Scattering
- Phosphorescence
- Interreflection



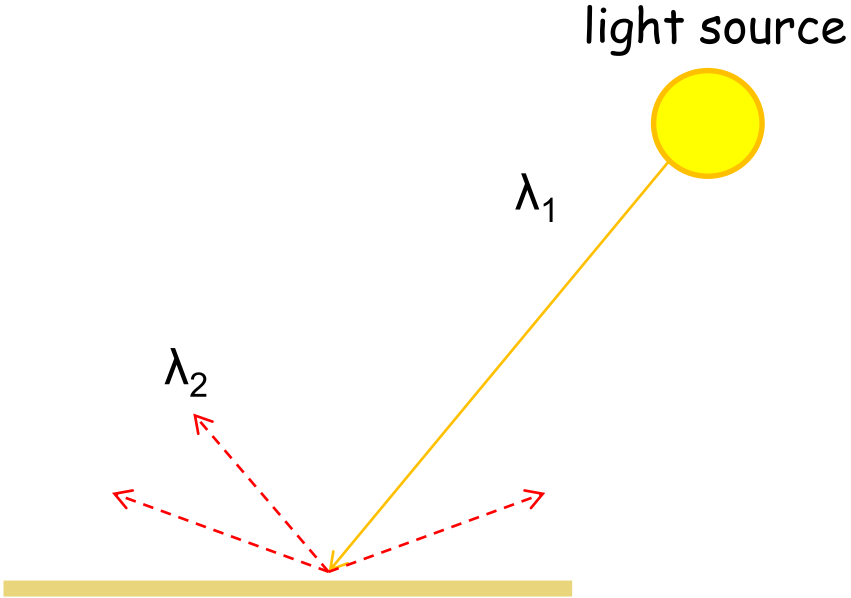
A photon's life choices

- Absorption
- Diffusion
- Reflection
- Transparency
- **Refraction**
- Fluorescence
- Scattering
- Phosphorescence
- Interreflection



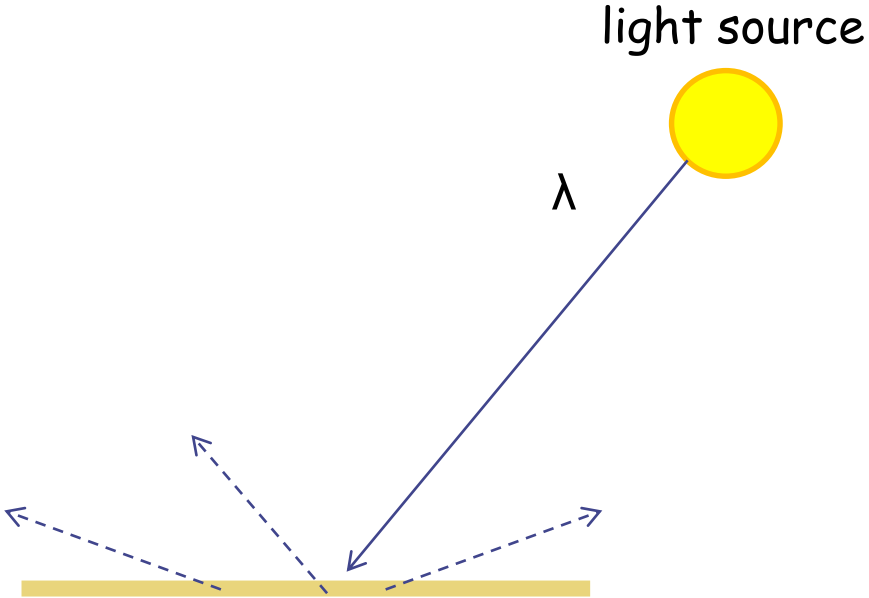
A photon's life choices

- Absorption
- Diffusion
- Reflection
- Transparency
- Refraction
- **Fluorescence**
- Scattering
- Phosphorescence
- Interreflection



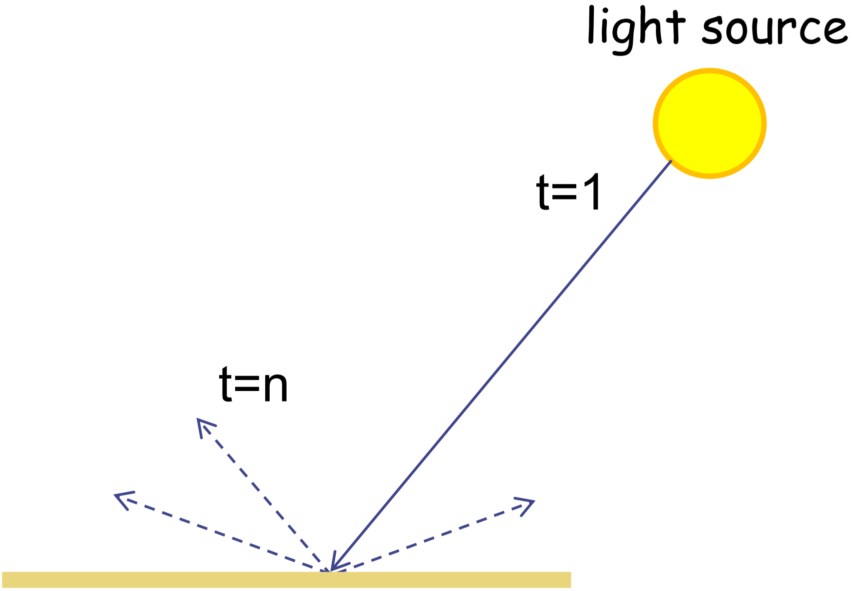
A photon's life choices

- Absorption
- Diffusion
- Reflection
- Transparency
- Refraction
- Fluorescence
- **Subsurface scattering**
- Phosphorescence
- Interreflection



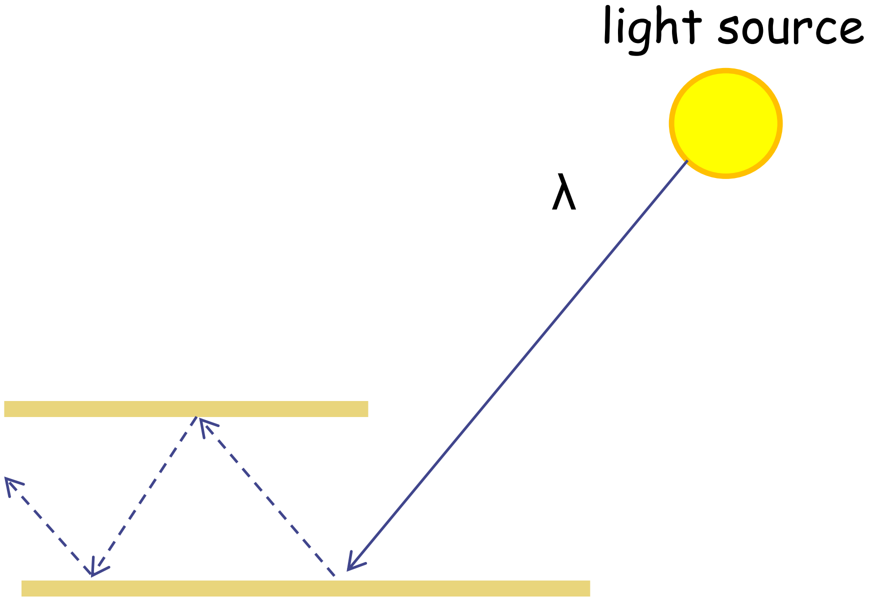
A photon's life choices

- Absorption
- Diffusion
- Reflection
- Transparency
- Refraction
- Fluorescence
- Scattering
- **Phosphorescence**
- Interreflection



A photon's life choices

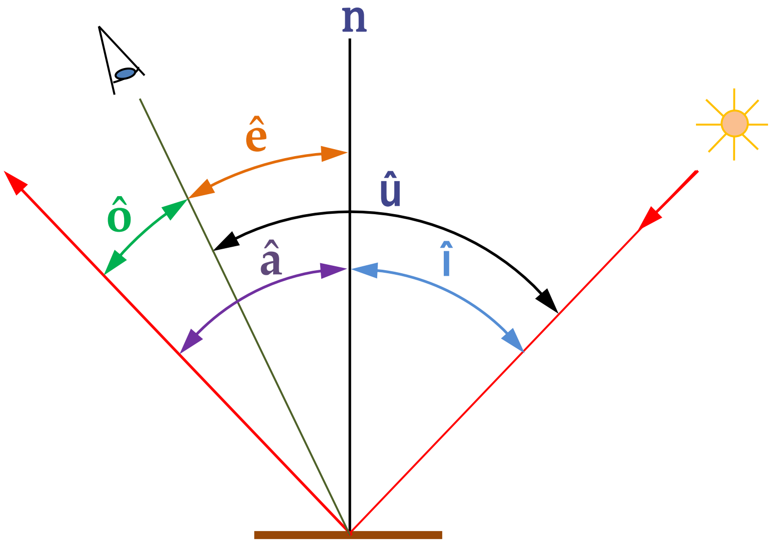
- Absorption
- Diffusion
- Reflection
- Transparency
- Refraction
- Fluorescence
- Scattering
- Phosphorescence
- **Inter-reflection**



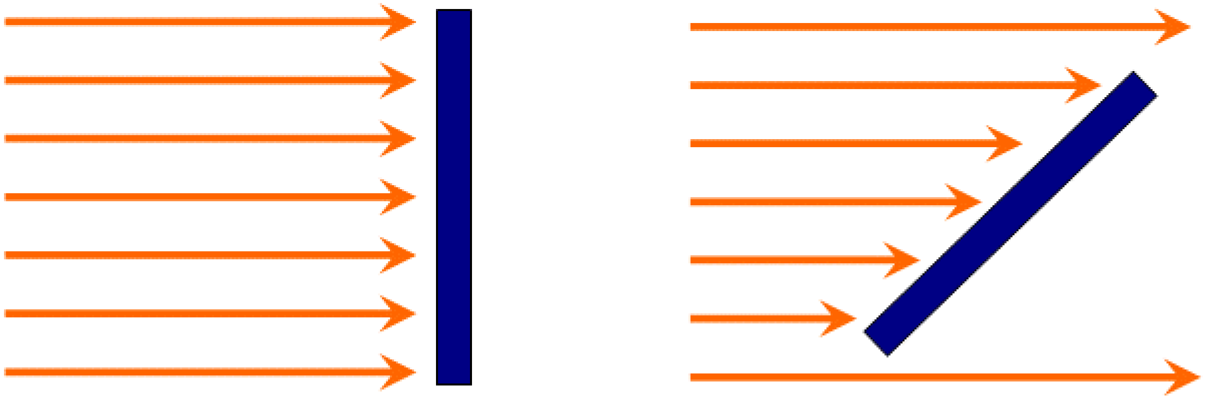
(Specular Interreflection)

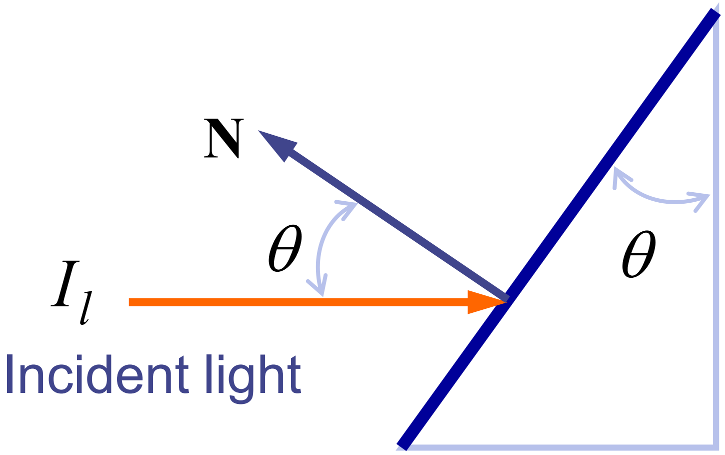
Local rendering

- Optical geometry of the light-source/eye-receiver system
- Notation:
  - n is the normal to the surface at the incidence point
  - î corresponds to the incidence angle
  - â = î is the reflectance angle
  - ô is the mirrored emergence angle
  - û is the phase angle
  - ê is the emergence angle.



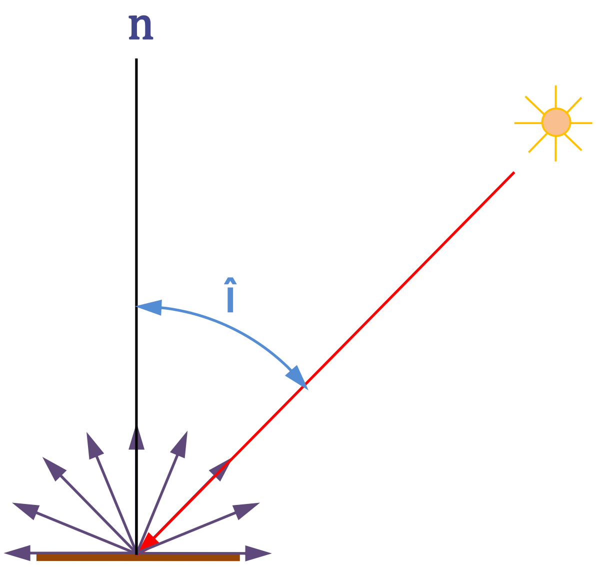
The amount of incident light depends on the orientation of the surface relative to the light source direction.

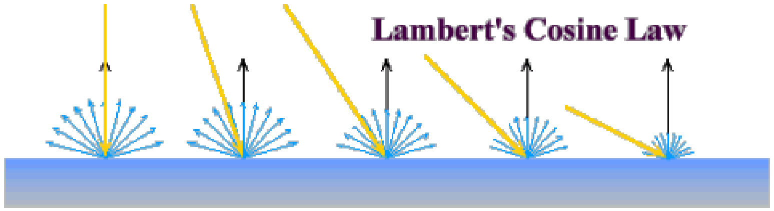




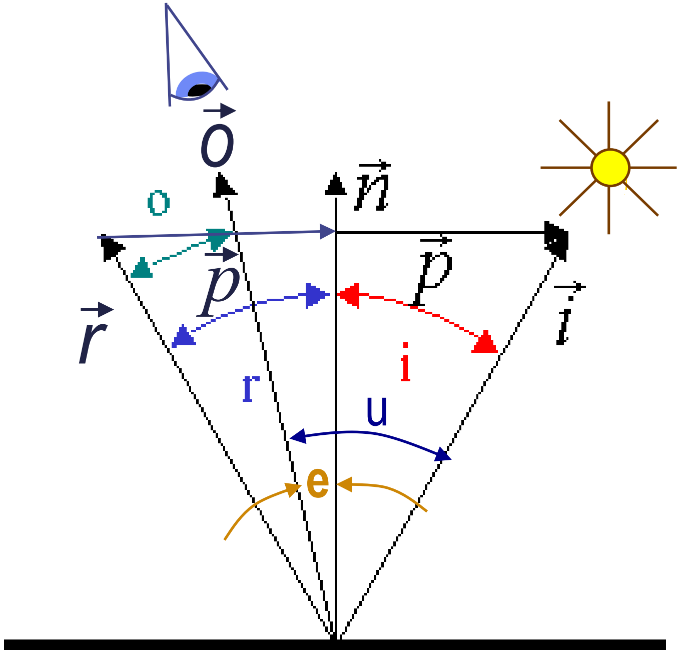
The Lambertian model

$$\Phi = \begin{cases} \cos \hat{\imath} & -\frac{\pi}{2} \leq \hat{\imath} \leq \frac{\pi}{2} \\ 0 & \textit{elsewhere} \end{cases}$$





The mirrored emergence angle



$$\|\vec{r}\| = \|\vec{o}\| = \|\vec{n}\| = \|\vec{i}\| = 1$$

$$(\vec{i}\,\vec{n})\vec{n} + \vec{p} = \vec{i} \rightarrow \vec{p} = \vec{i} - (\vec{i}\,\vec{n})\vec{n}$$

$$\vec{r} = \vec{i} - 2\vec{p} = 2(\vec{i}\,\vec{n})\vec{n} - \vec{i}$$

$$\vec{r}\vec{o} = 2(\vec{i}\,\vec{n})(\vec{n}\vec{o}) - (\vec{i}\,\vec{o})$$

$\cos(\mathbf{e}) = \vec{n}\vec{o}$

$\cos(\mathrm{o}) = \vec{o}\vec{r}$

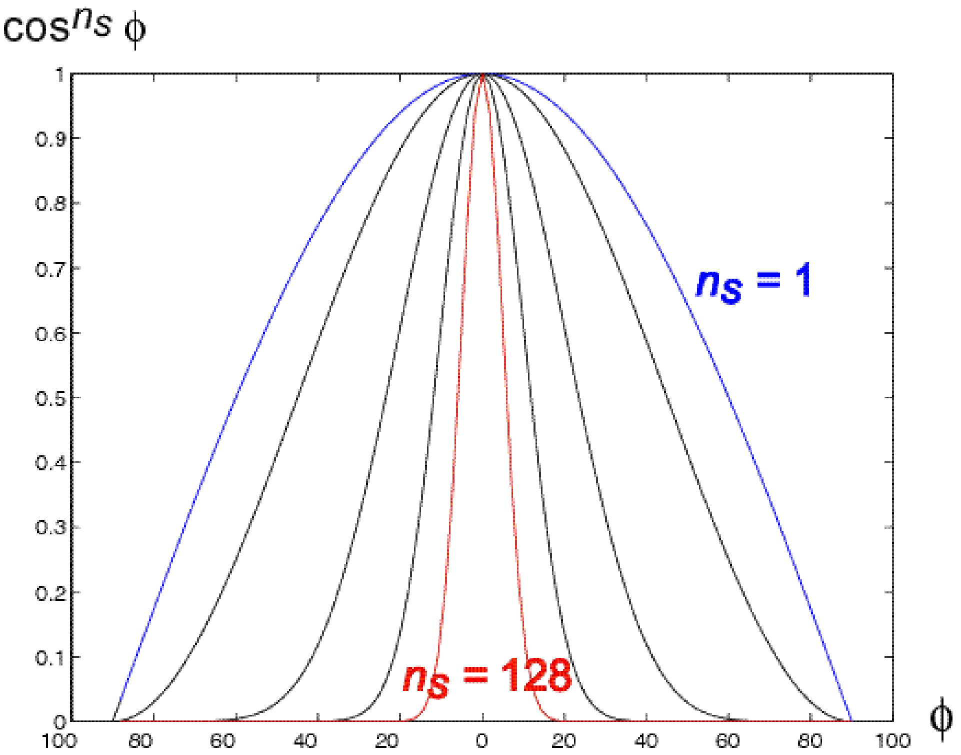
$\cos(\mathbf{u}) = \vec{o}\vec{i}$

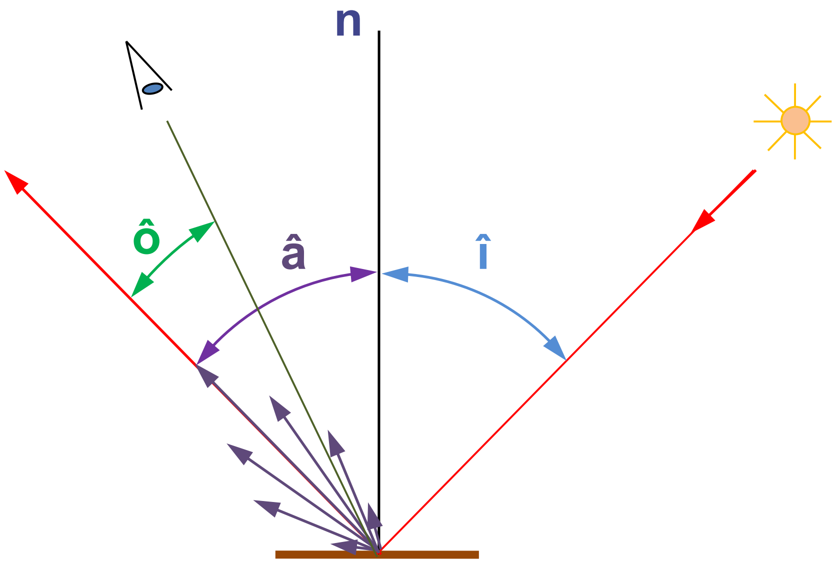
$\cos(\mathrm{i}) = \vec{i}\,\vec{n}$

cos (o) = 2 cos (i) cos (e) - cos (u)

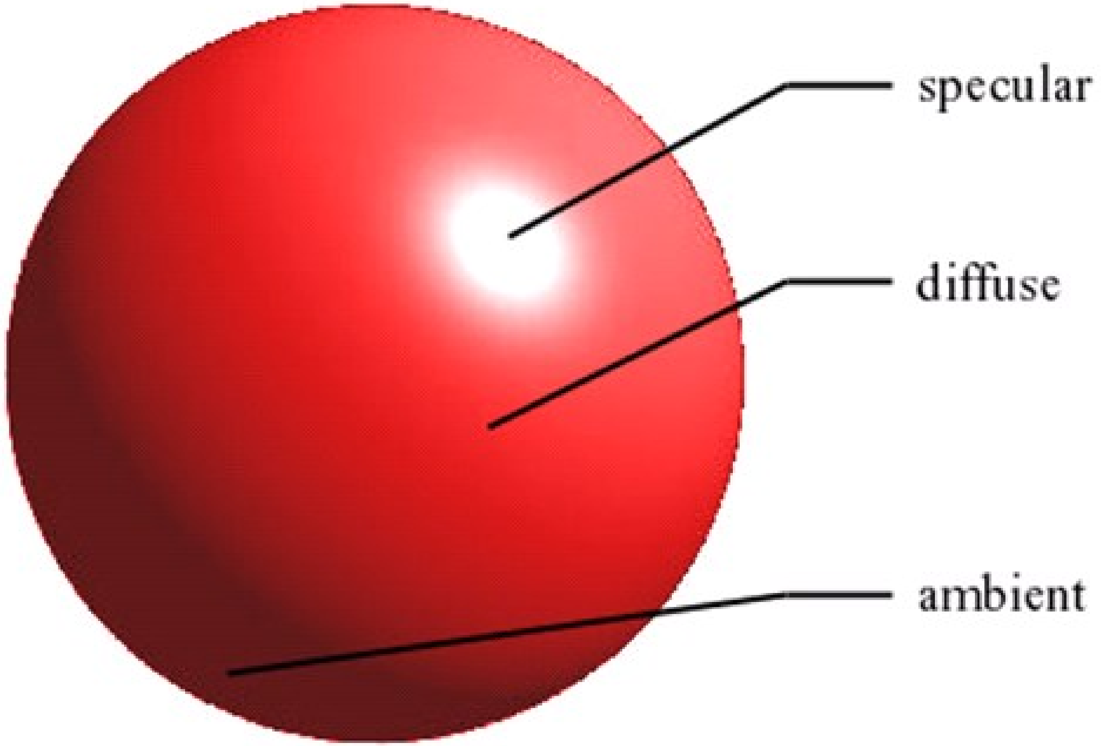
The specular model

$$\Phi = \begin{cases} \mathbf{cos}^m \hat{o} & -\frac{\pi}{2} \le \hat{o} \le \frac{\pi}{2} \\ 0 & elsewhere \end{cases}$$





Components of Reflections



Ambient – surface exposed to **indirect light** reflected from nearby objects.

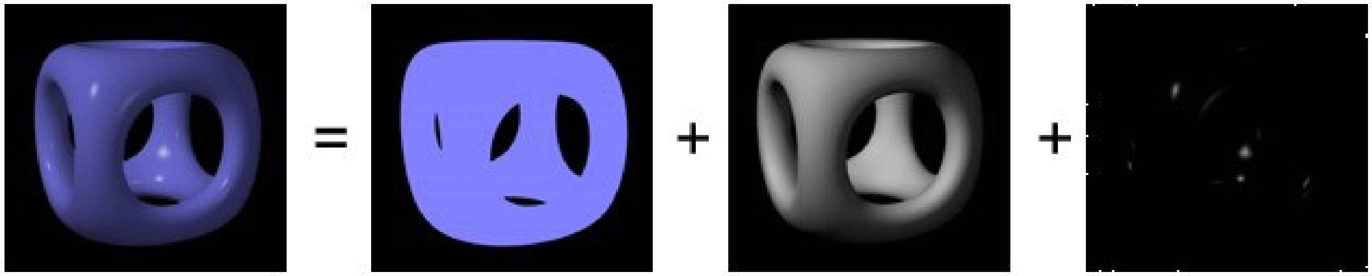
Diffuse – reflection from **incident** light with equal intensity in all directions. Depends on surface properties.

Specular – near **total** of the incident light around reflection angle.

The Phong model

- *a* is the amount of incident light diffused according to a Lambertian model (isotropic) independent from the receiver's position
- *b* is the amount of incident light specularly reflected by the object, which depends on the phase angle, and m being the exponential specular reflection coefficient
- *c* accounts for the background illumination

$$\Phi = a\cos\hat{\imath} + b\cos^{m}\hat{o} + c$$



color and ambient

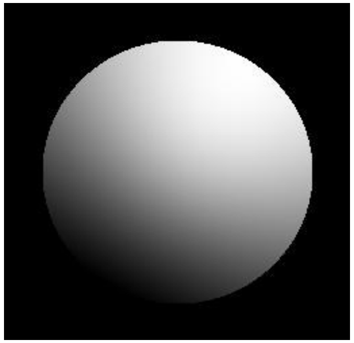
diffuse

specularity

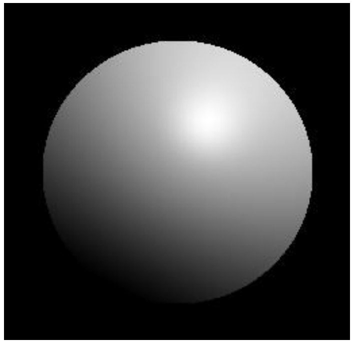
Bùi Tường Phong, Illumination for computer generated images, Commun. ACM 18 (6) (1975) 311317.

Sphere (m=10, c=0)

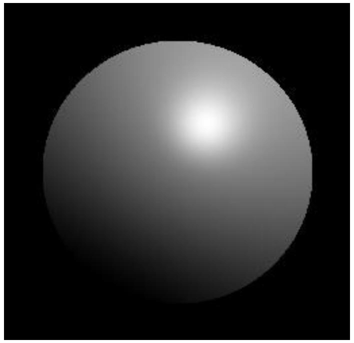
a=1.0 b=0.0



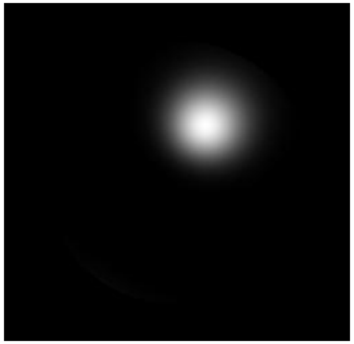
0.8 0.2



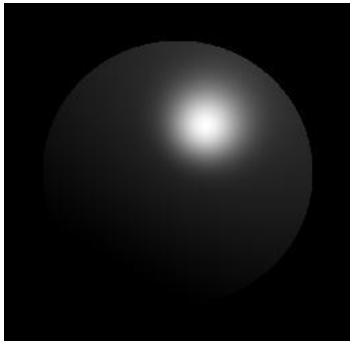
0.6 0.4



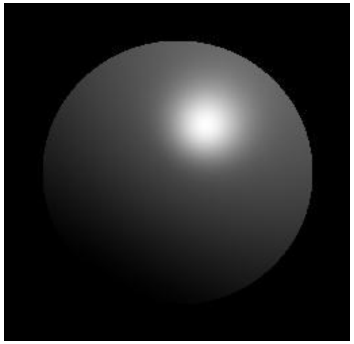
0.0 1.0



0.2 0.8

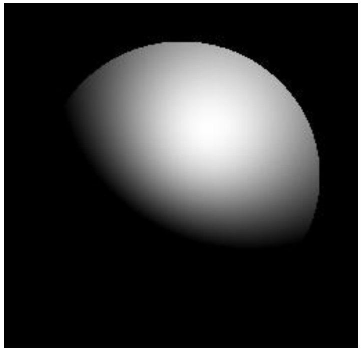


0.4 0.6

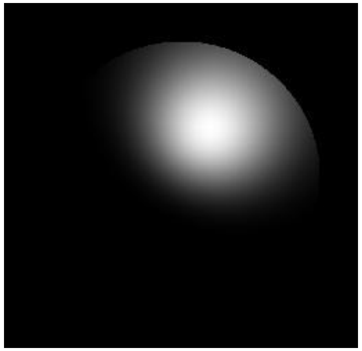


Specular sphere (b=0.9, c=0.1)

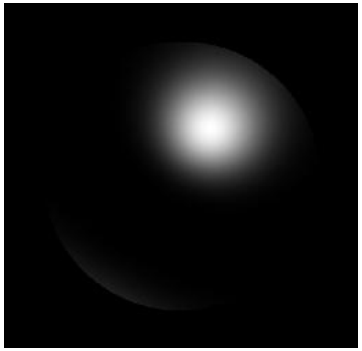
m=1



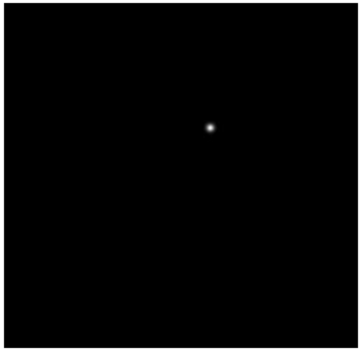
m=3



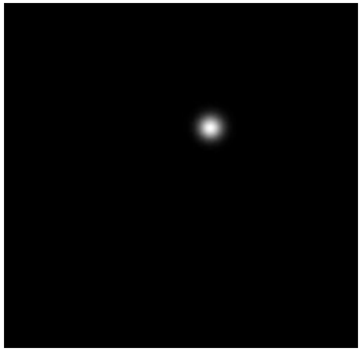
m=6



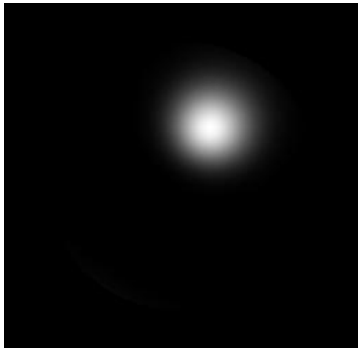
m=1000



m=100

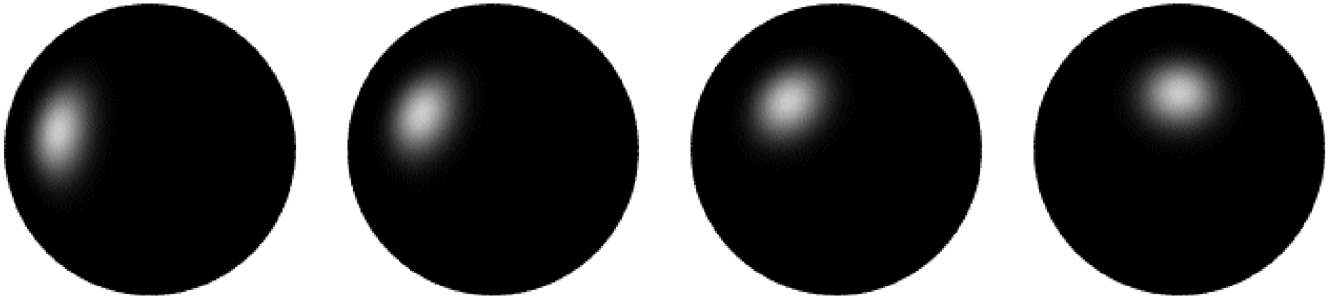


m=10

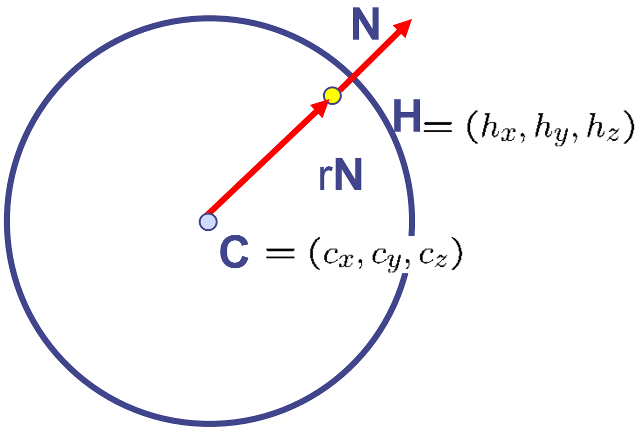


Computing light source directions

- Trick: place a chrome sphere in the scene



  - the location of the highlight tells you where the light source is



$$\|H - C\| = r$$

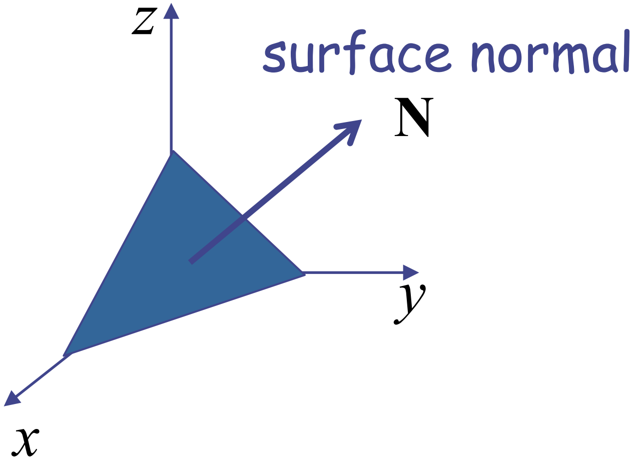
Gradient space

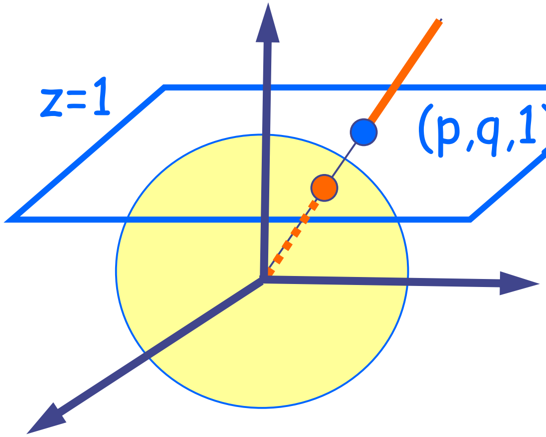
$$z = f(x, y)$$

$$p = -\frac{\partial f}{\partial x} \qquad q = -\frac{\partial f}{\partial y}$$

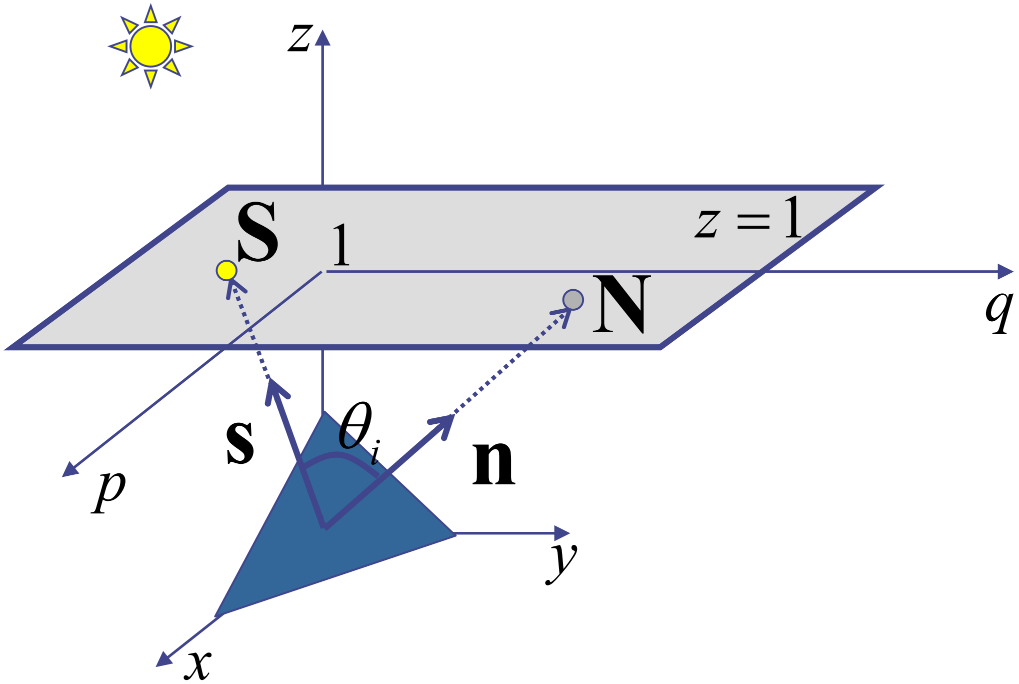
$$\mathbf{n} = (p, q, 1)$$

$$\hat{\mathbf{n}} = \frac{1}{\sqrt{p^2 + q^2 + 1}}(p, q, 1)$$





Gradient Space



Normal vector

$$\mathbf{n} = \frac{\mathbf{N}}{|\mathbf{N}|} = \frac{(p,q,1)}{\sqrt{p^2+q^2+1}}$$

Source vector

$$\mathbf{s} = \frac{\mathbf{S}}{|\mathbf{S}|} = \frac{(p_S,q_S,1)}{\sqrt{p_S^2+q_S^2+1}}$$

$$\cos\theta_i = \mathbf{n}\cdot\mathbf{s} = \frac{(pp_S+qq_S+1)}{\sqrt{p^2+q^2+1}\sqrt{p_S^2+q_S^2+1}}$$

$z=1$ plane is called the Gradient Space (*p q* plane)

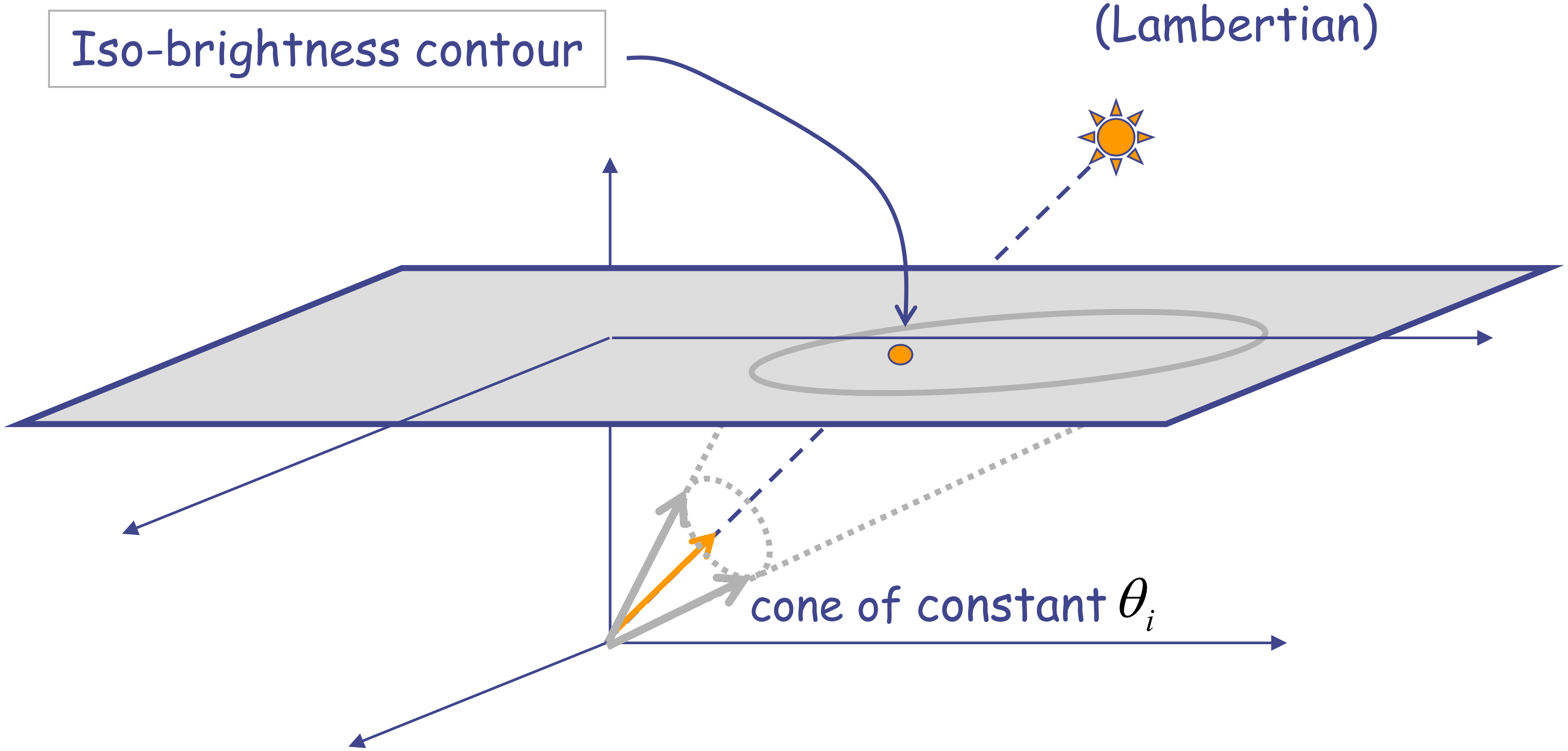
Every point on it corresponds to a particular surface orientation

Reflectance Map

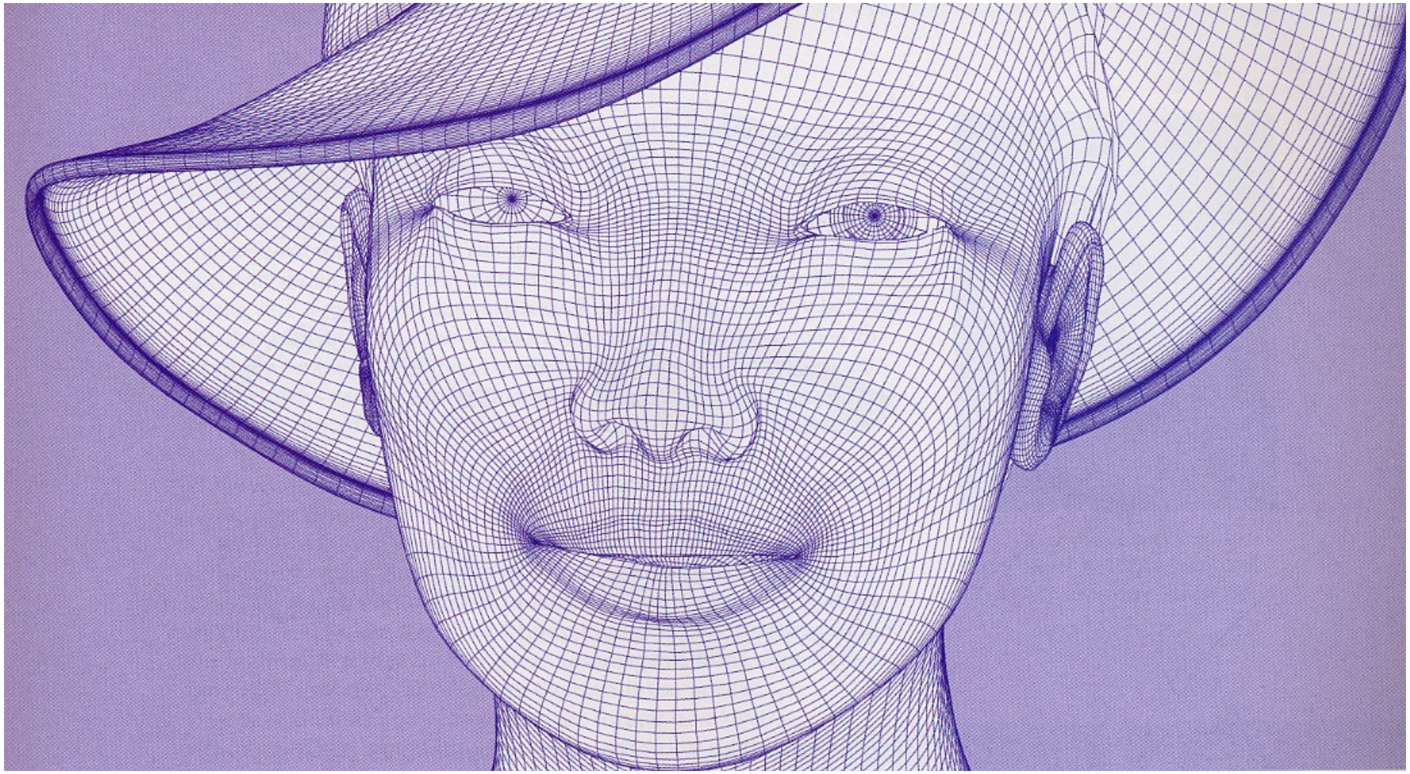
- Lambertian case

$$I = \cos\theta_i = \mathbf{n}\cdot\mathbf{s} = \frac{(pp_s + qq_s + 1)}{\sqrt{p^2+q^2+1}\sqrt{p_S^2+q_S^2+1}} = \boxed{R(p,q)}$$

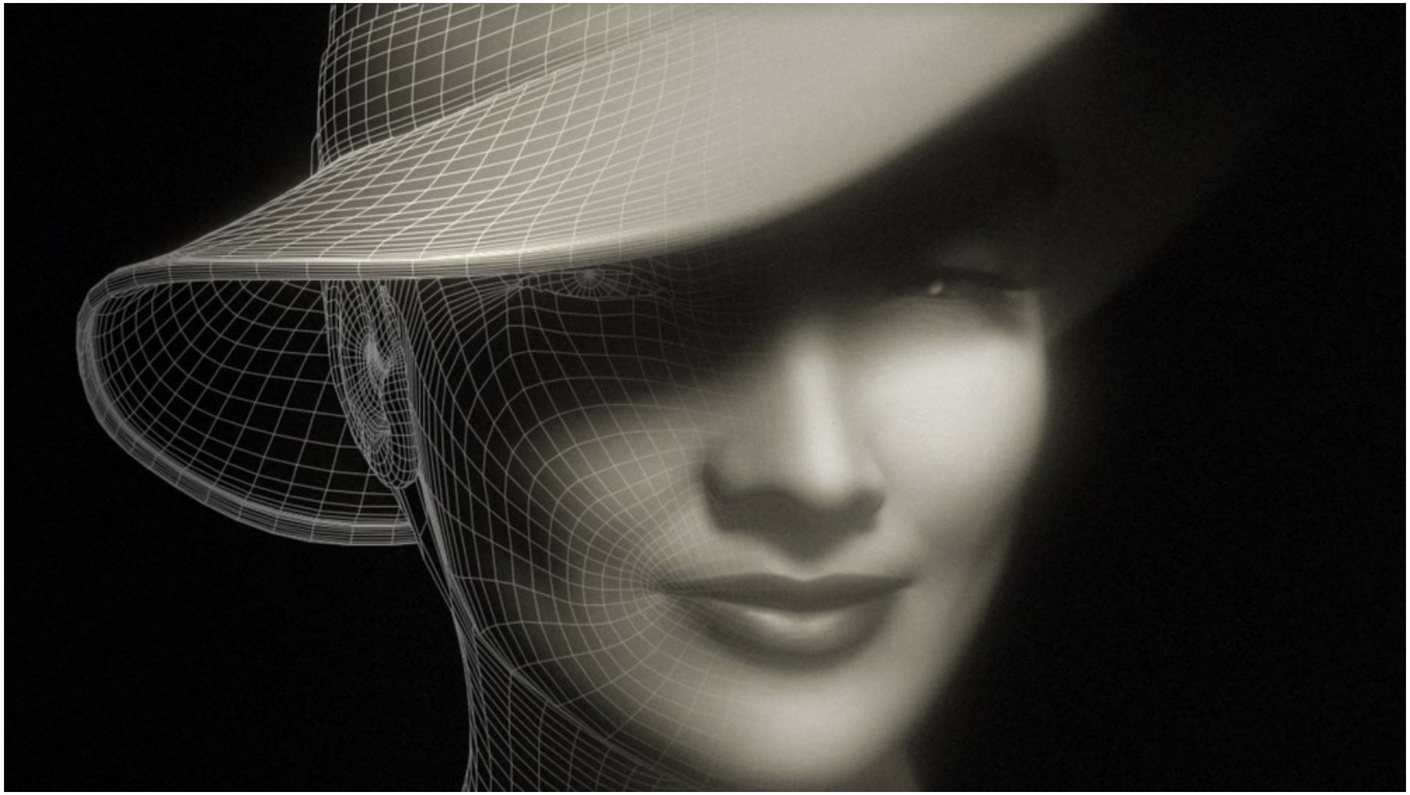
Reflectance Map (Lambertian)



Wire frame surface presentation

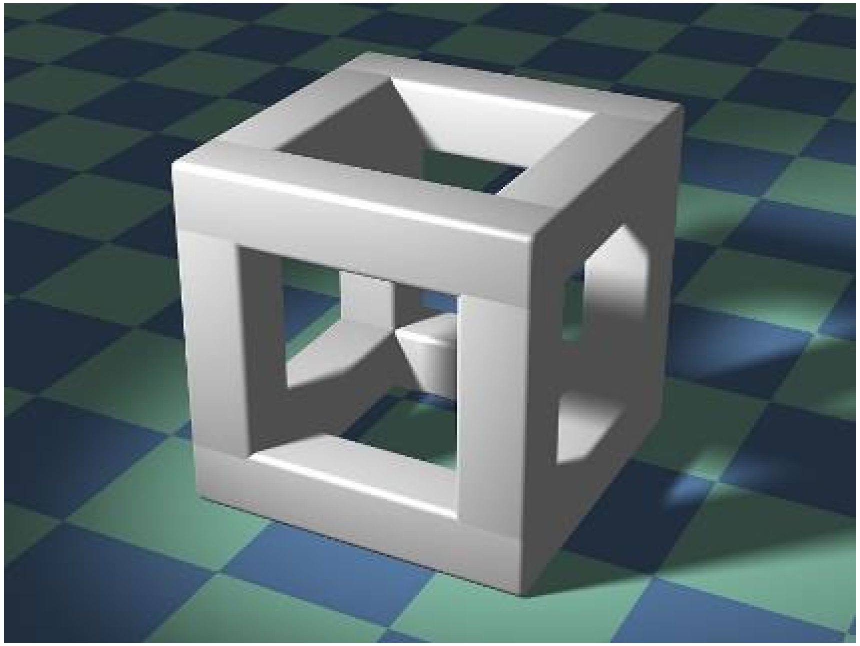


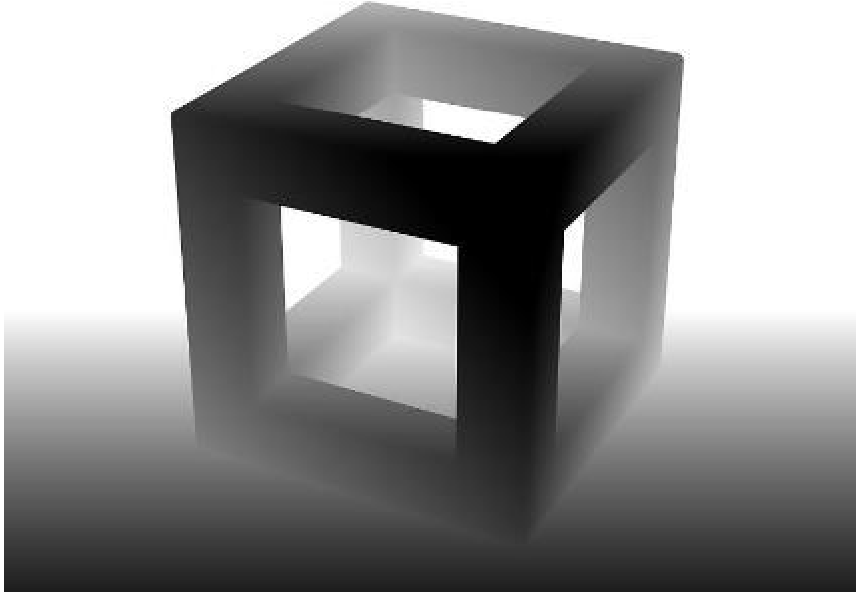
Local rendering



http://www.danielrobichaud.com/animation/#/marlene/

Range finder





Reflectance maps

- Let us take a reference system where the optical axis of the acquisition system (the receiver) coincides with the z axis.
- The surface described by the function z = f(x, y) has the normal vector: $(\partial z/\partial x, \partial z/\partial y, -1)^t$.
- Calling $p=\partial z/\partial x$ and $q=\partial z/\partial y$ there is a one-to-one correspondence between the plane p, q (called *gradient plane)* and the normal directions to the surface.
- The three angles î, û and ê may be computed with the following formulas:

$$\cos \hat{e} = \frac{1}{\sqrt{1+p^2+q^2}}$$

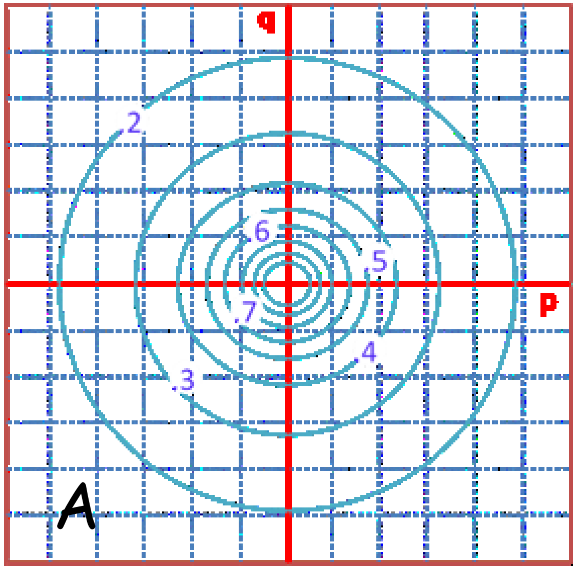
$$\cos \hat{\imath} = \frac{1+pp_s+qq_s}{\sqrt{1+p^2+q^2}\sqrt{1+p_s^2+q_s^2}} \qquad \cos \hat{u} = \frac{1}{\sqrt{1+p_s^2+q_s^2}}$$

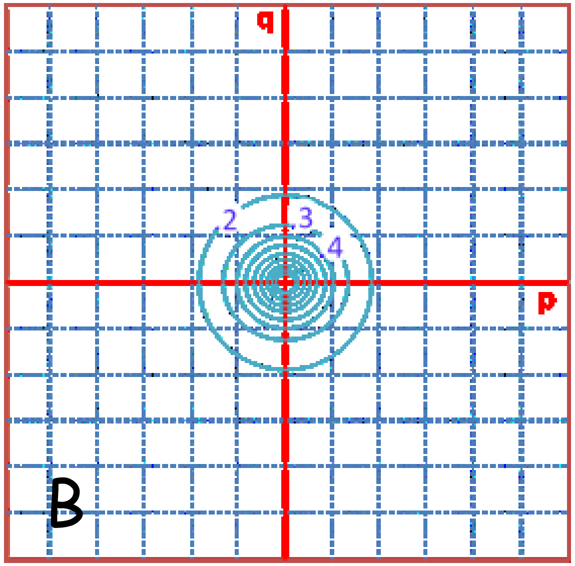
Reflectance maps examples

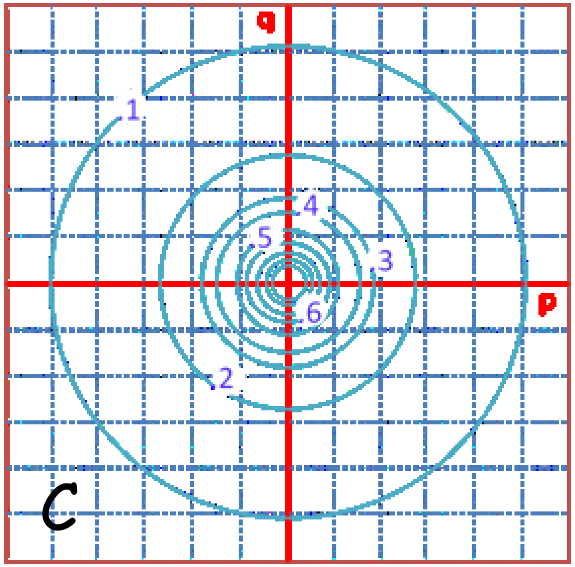
A. reflectivity map for a Lambertian case having both camera and light source coincident with the z-axis (0, 0, -1);

B. the specular case having the specularity index m=3;

C. an intermediate case with a = b = 0.5, m=3 and c = 0

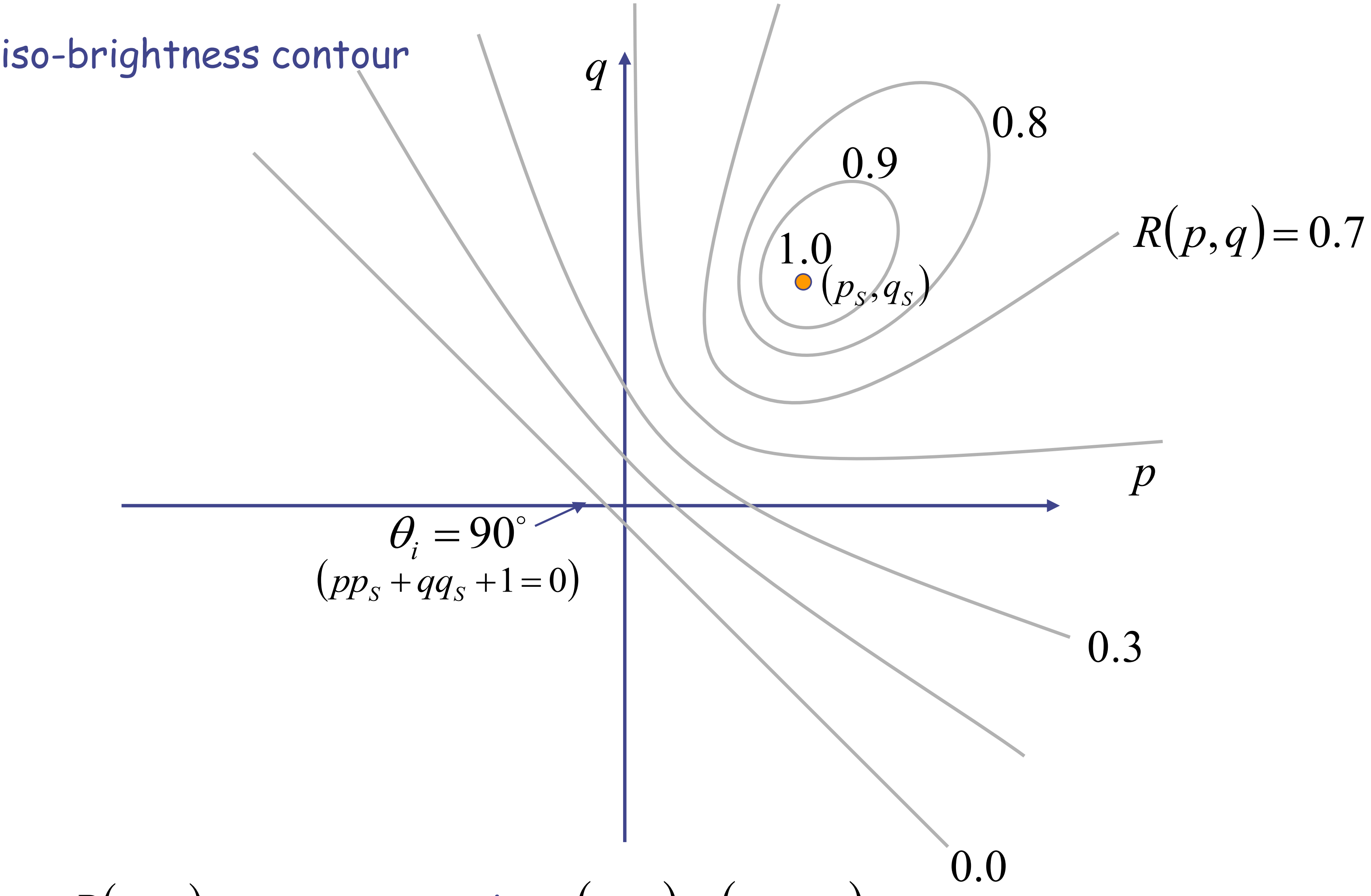






Reflectance Map

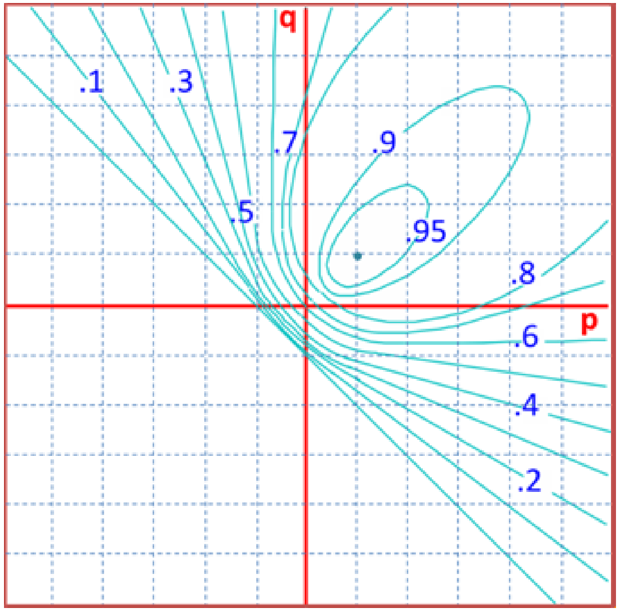
- Lambertian case

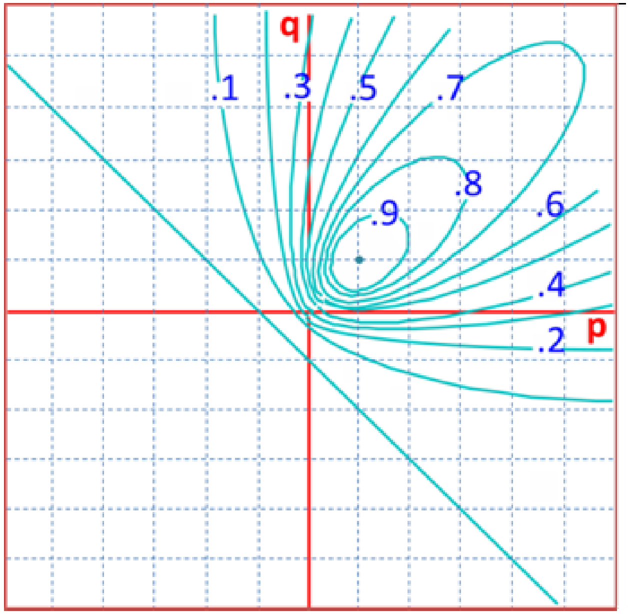


Note: $R(p,q)$ is maximum when $(p,q) = (p_S, q_S)$

Reflectance maps

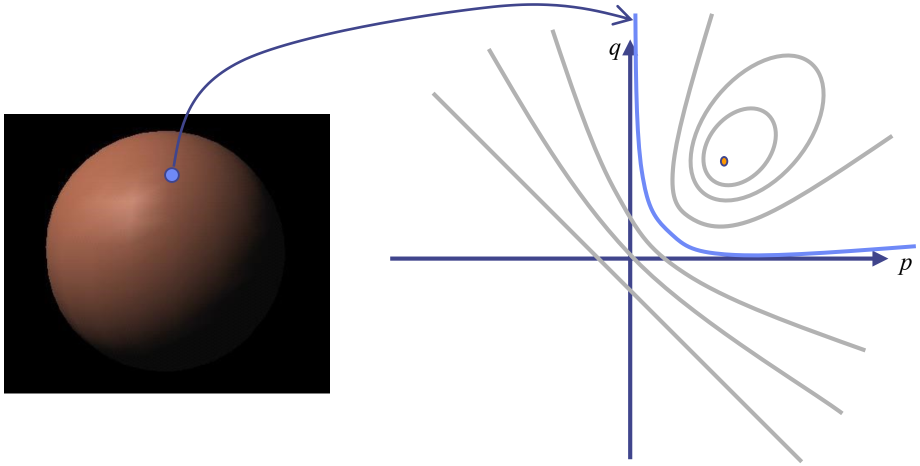
- Reflectivity map for a Lambertian model; having the camera on the z-axis (0,0,-1) and the light source positioned at (1, 1, -1). The isoluminance patterns are quadrics labelled with their corresponding ratios, the incident light source corresponds to the bisector of the first octant in 3D space.
- Reflectivity map for a specular model having the specularity index m = 3.





Shape from a Single Image?

- Given a single image of an object with known surface reflectance taken under a known light source, can we recover the shape of the object?

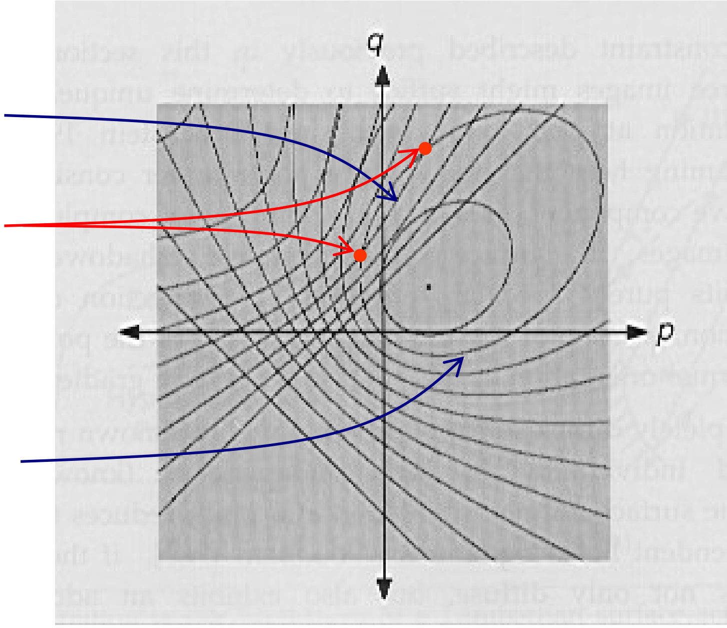


- Two reflectance maps?

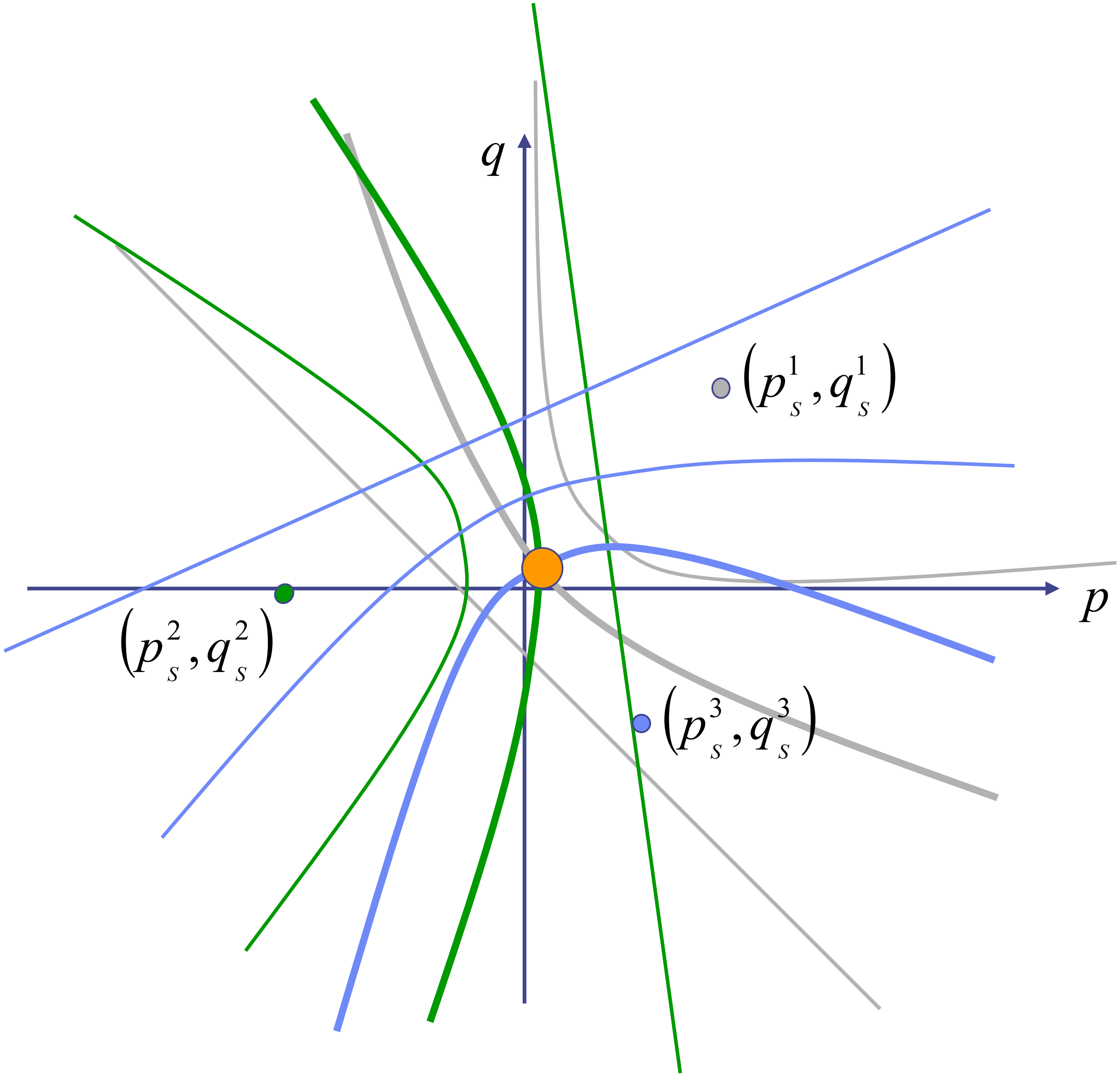
$R_2(p,q)=a$

Intersections:
2 solutions for
p and q.

$R_1(p,q)=b$

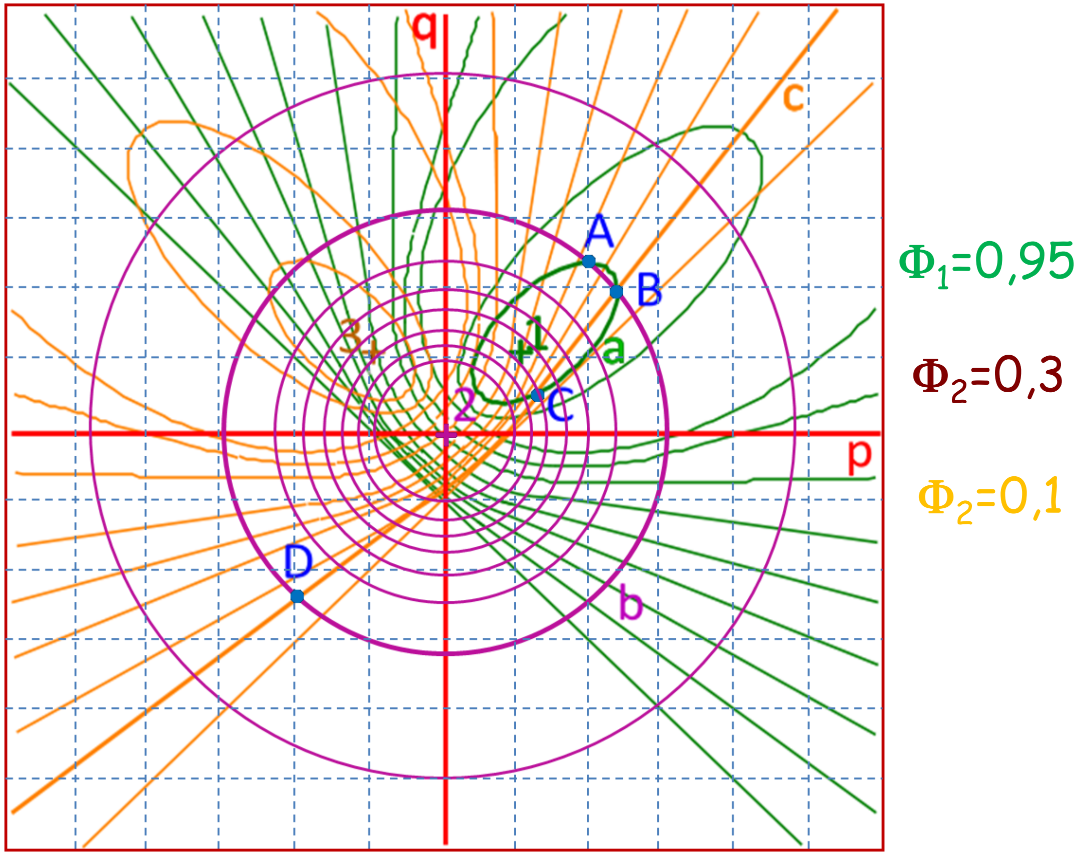


Photometric Stereo


q
p
$\left(p_s^1, q_s^1\right)$
$\left(p_s^2, q_s^2\right)$
$\left(p_s^3, q_s^3\right)$

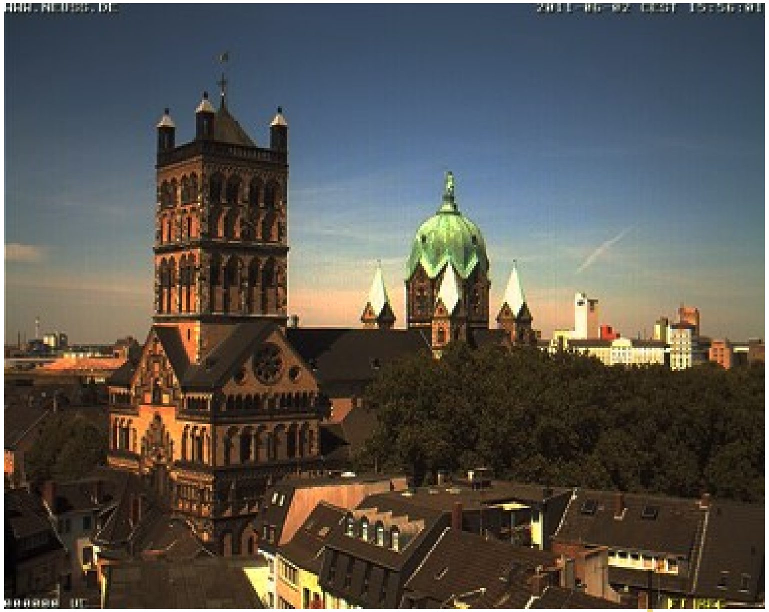
Photometric analysis

- Overlapped isoluminance patterns, for the Lambertian model, with three different positions of the light source used [(1,1,-1), (0,0,-1), (-1, 1,-1)] for determining the attitude of an object's facet.

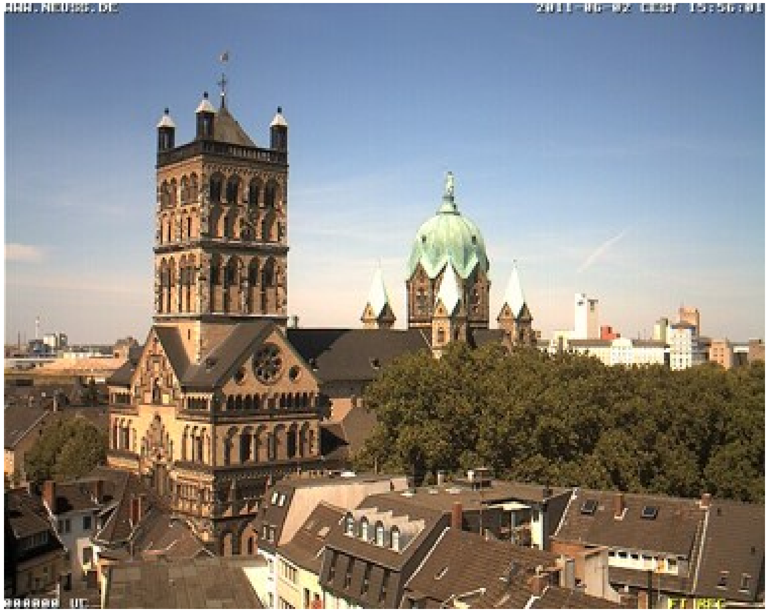


Radiometric Camera Calibration

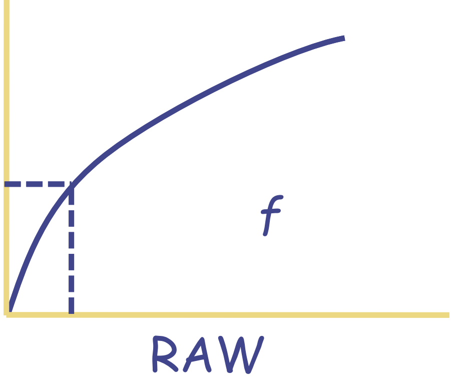
- Pixel intensities are usually *not* proportional to the energy that hit the CCD



RAW image



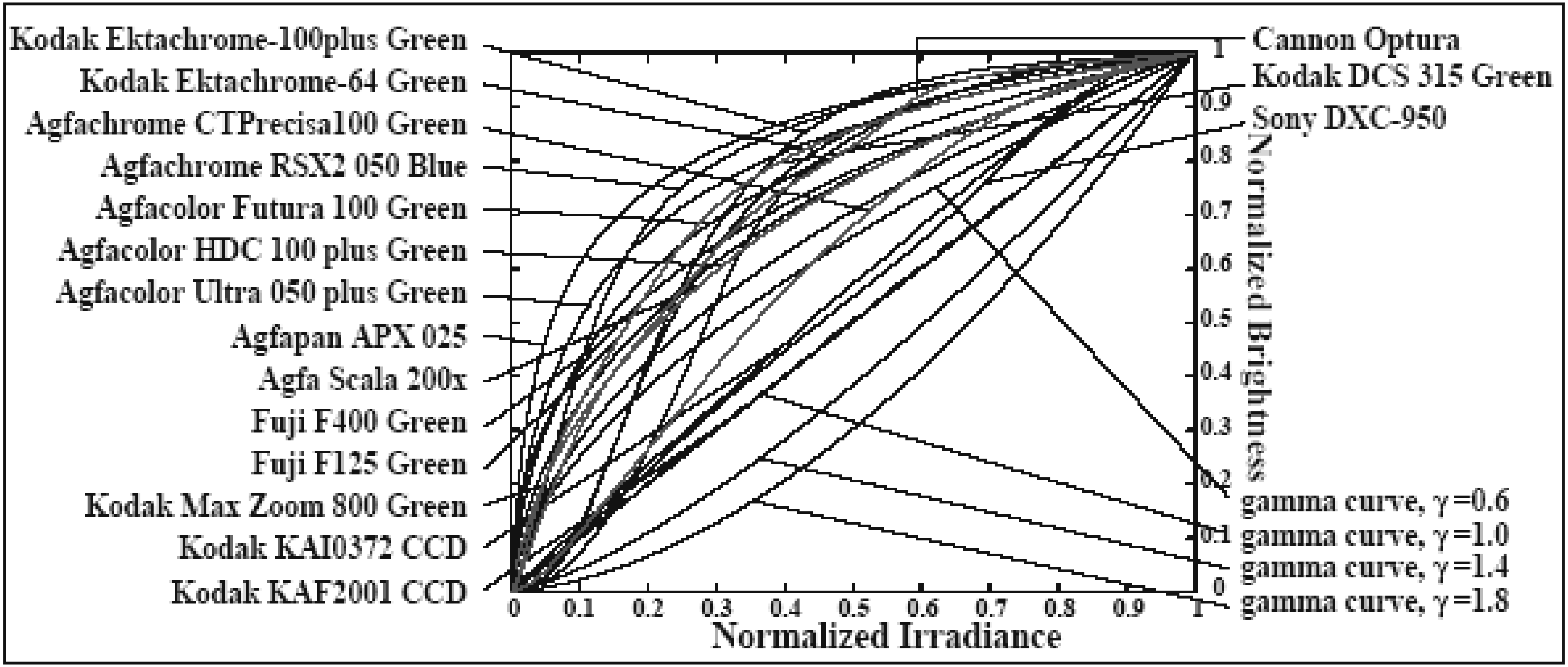
Published image



Austin Abrams

Radiometric Camera Calibration

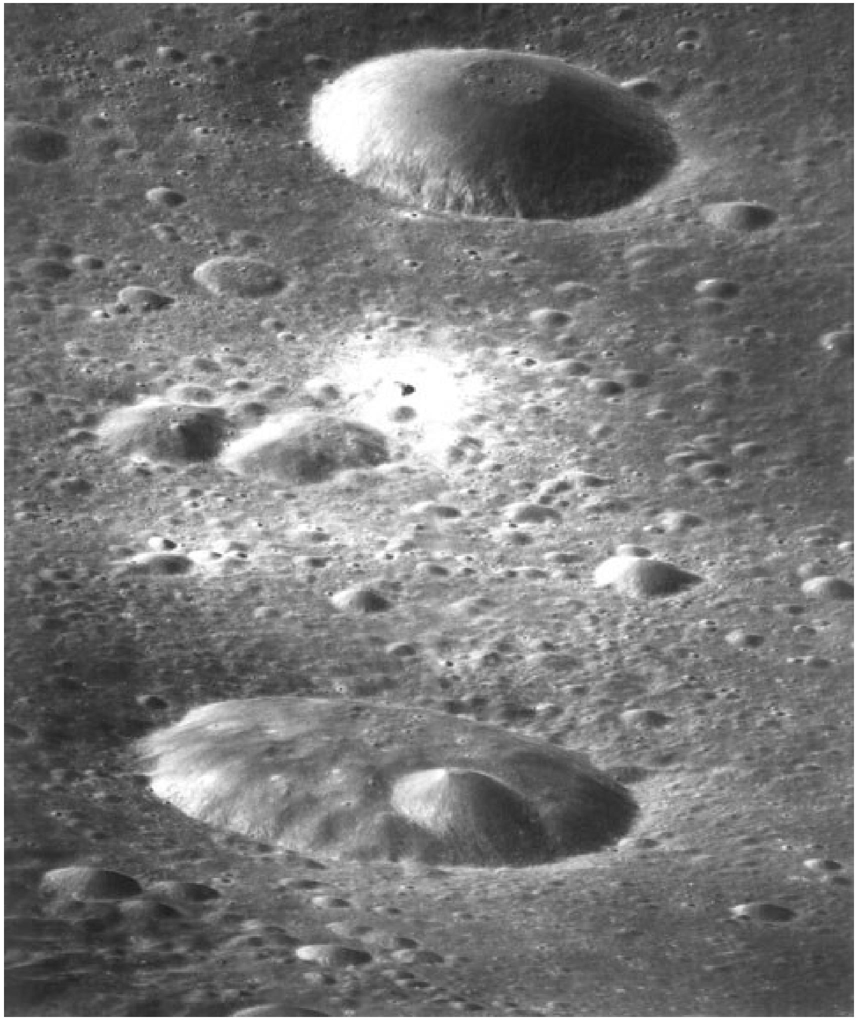
Observed = *f*(RAW)

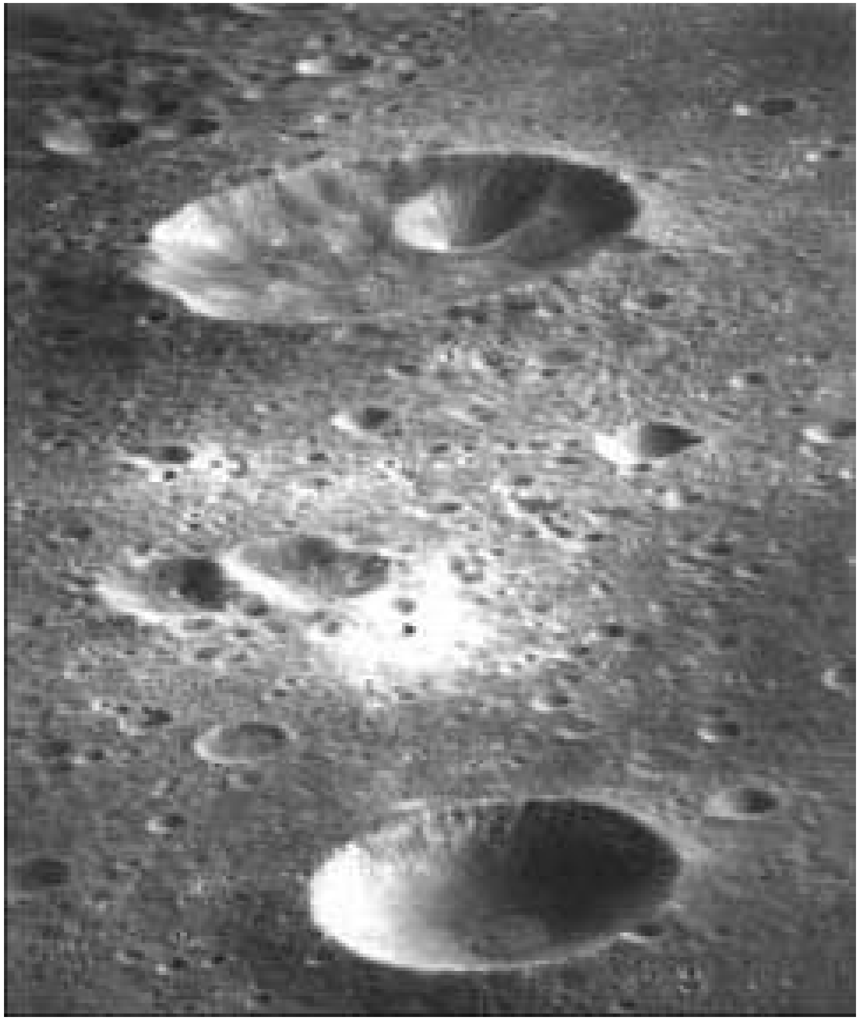


f^{-1} (Observed) = RAW

(Grossberg and Nayar)

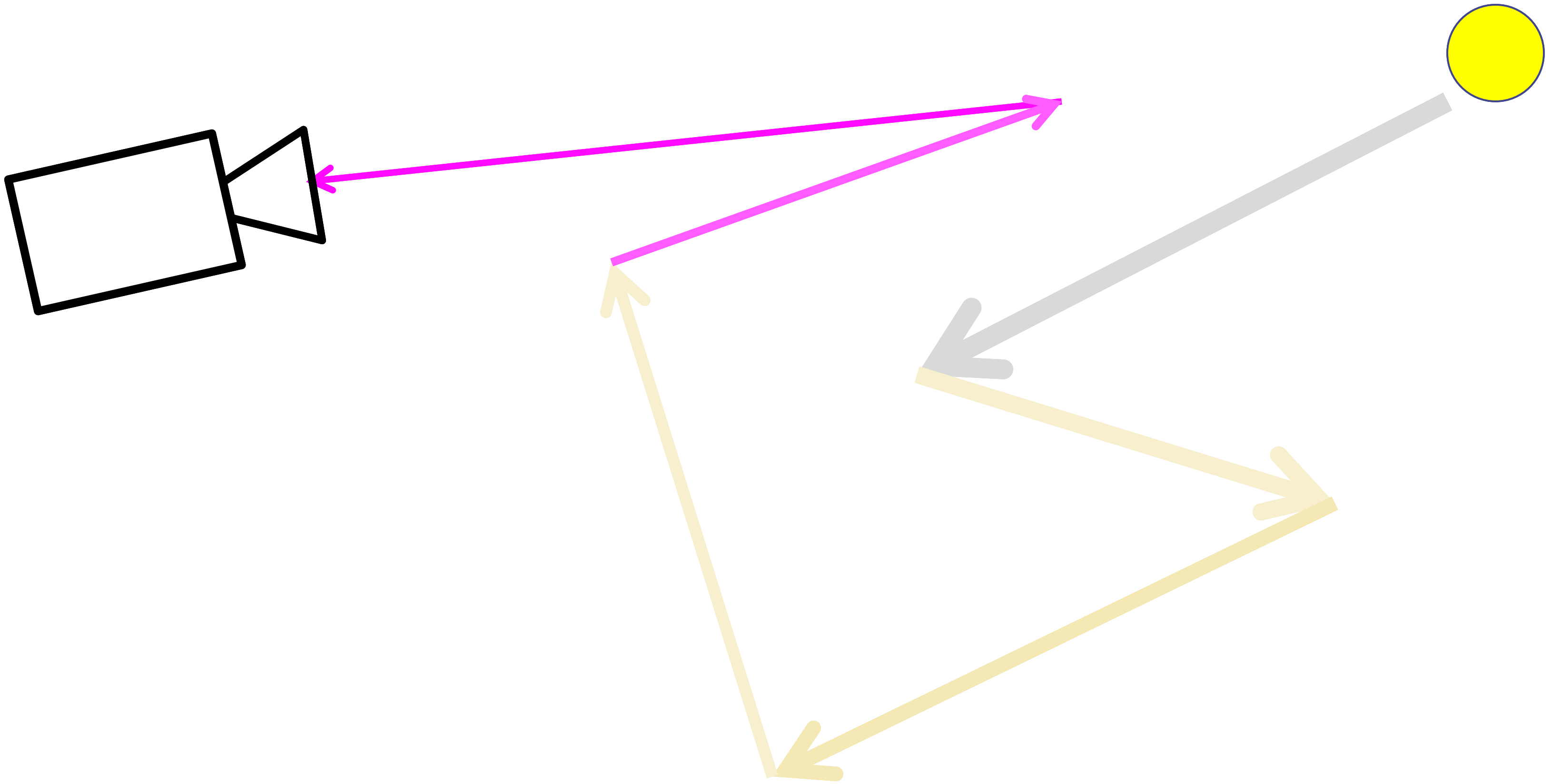
Vision is Inferential





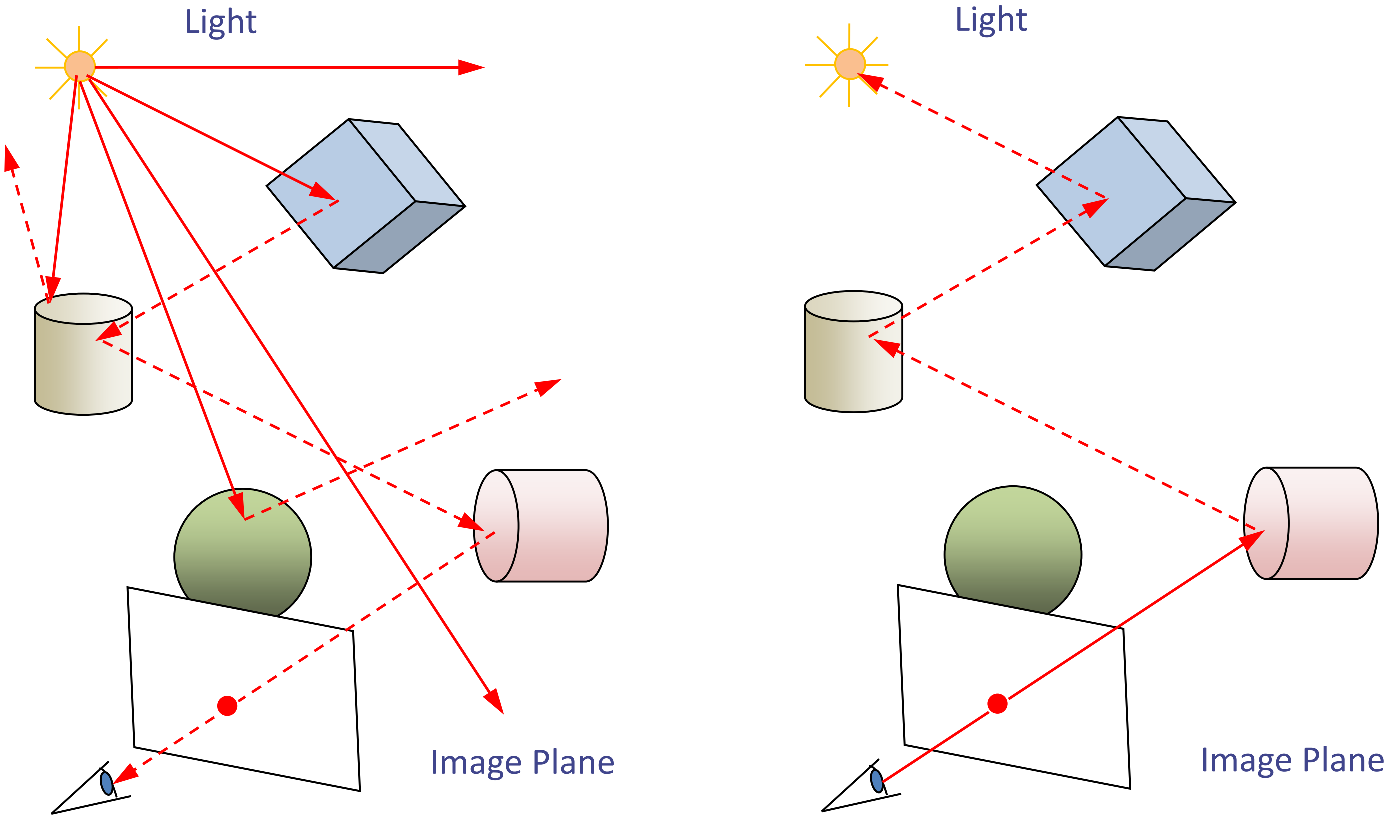
Prior Knowledge

The life of a photon



Global rendering: ray tracing

- The two basic schemes of forward (source to eye) and backward (eye to source) ray tracing, this last is computationally efficient but and cannot properly model diffuse reflections and all other changes in light intensity due to non-specular reflections



Global rendering: ray tracing

- A ray sent from the eye to the scene through the image plane may intersect an object; in this case secondary rays are sent out and three different lighting models can be considered:
  - *transmission*. The secondary ray is sent in the direction of refraction, following the Descartes'law;
  - *reflection*. The secondary ray is sent in the direction of reflection, and the Phong model is applied;
  - *shadowing*. The secondary ray is sent toward a light source, if intercepted by an object the point is shadowed.

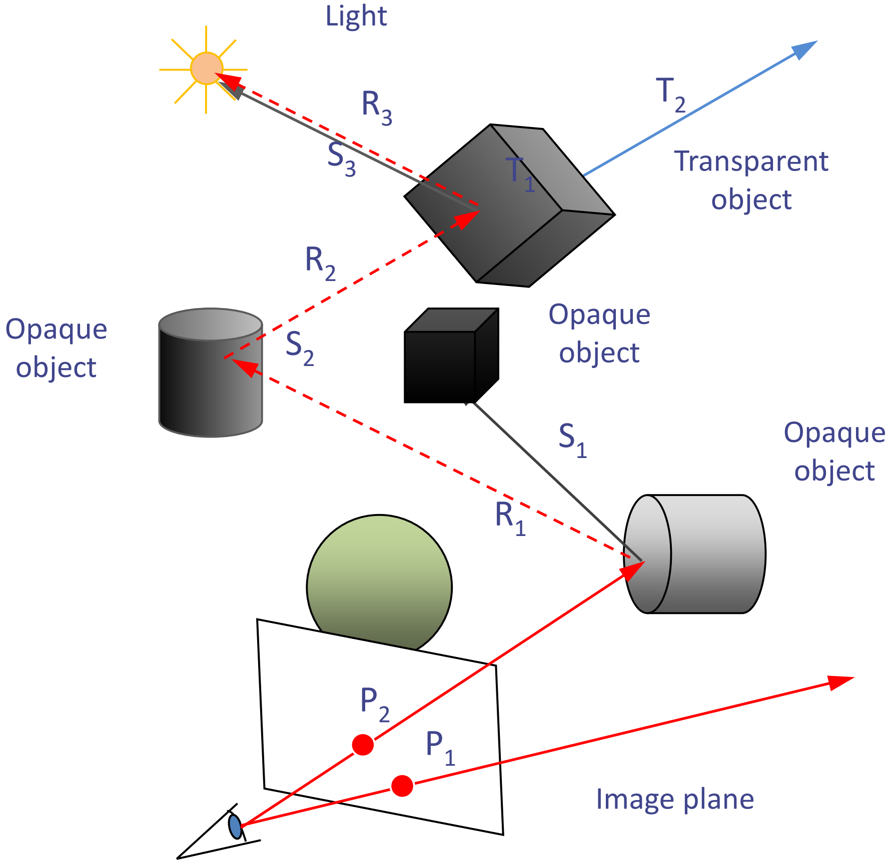
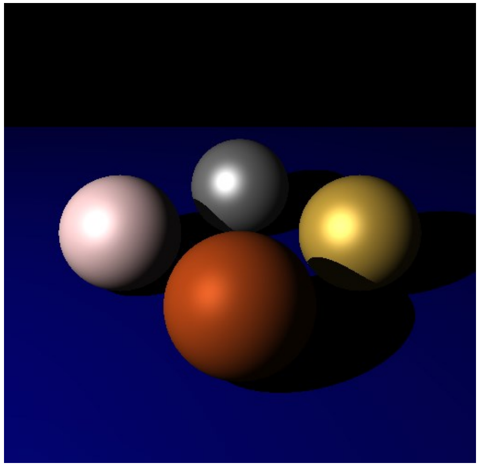
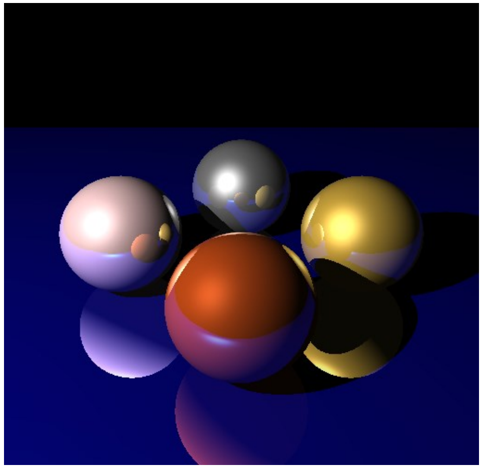


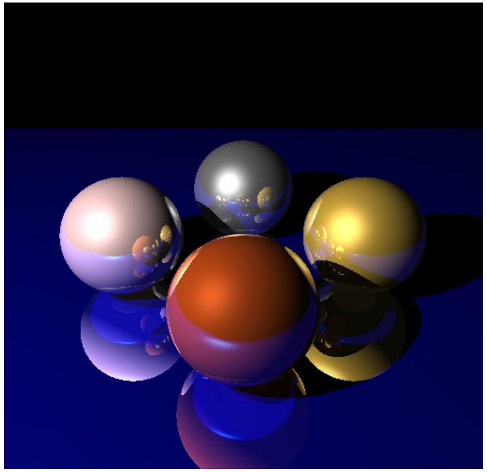
Figure. A background pixel corresponding to the ray P_1 and a pixel representing an object (ray P_2). The secondary rays triggered by the primary ray P_2 according to the three models: transmission (T_1 and T_2), reflection (R_1, R_2 and R_3), and tentative shadowing (S_1, S_2: true shadows, and S_3).

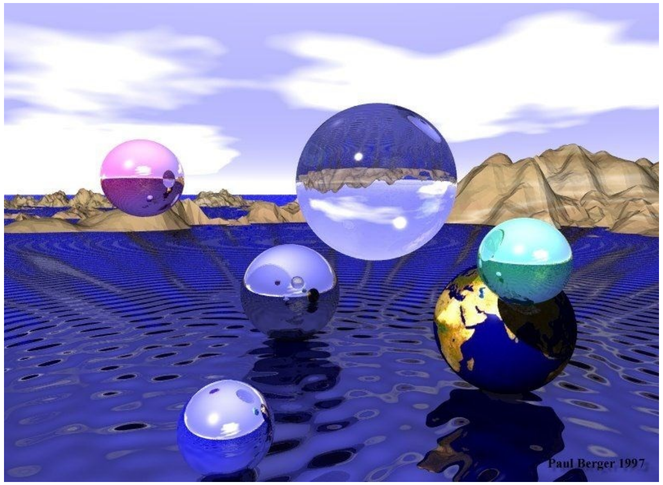
Reflections and transparencies



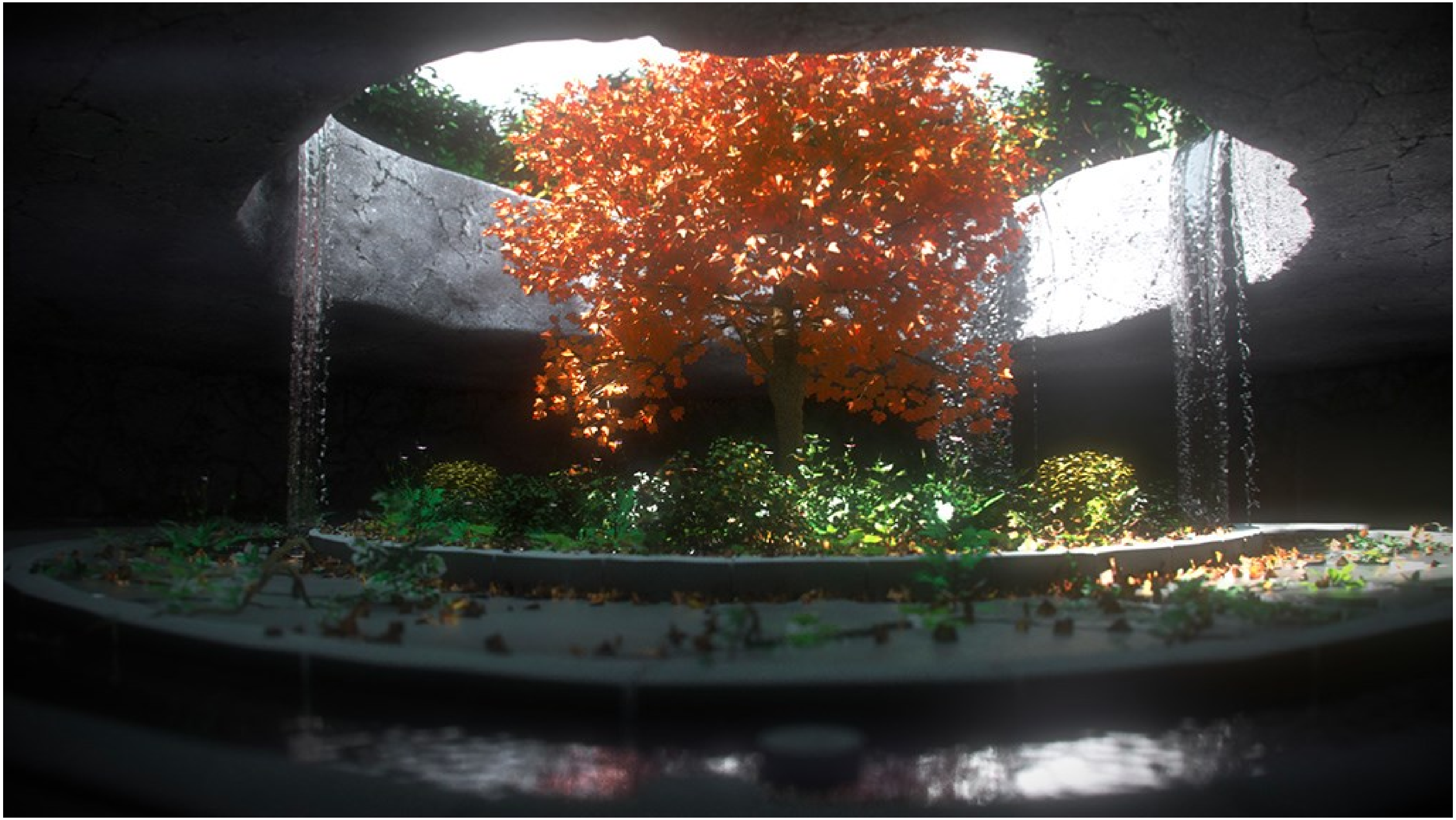


Created by David Derman – CISC 440





Example



Global rendering: radiosity

- The **radiosity** method has been developed to model the diffuse-diffuse interactions so gaining a more realistic visualization of surfaces.

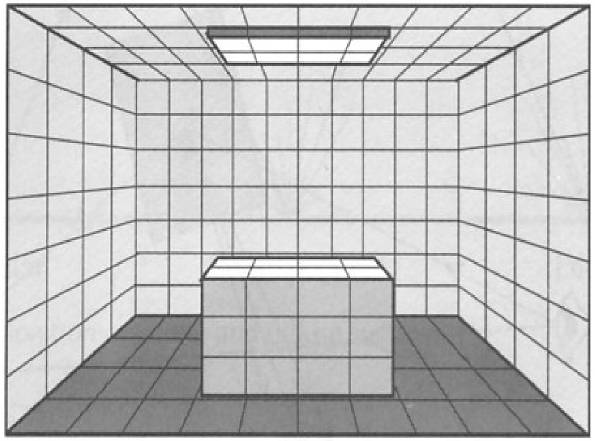
- The diffused surfaces scatter light in all directions (i.e. in Lambertian way). Thus a scene is divided into *patches* - small flat polygons. For each patch the goal is to measure energies emitted from and reflected to respectively. The radiosity of the patch *i* is given by:

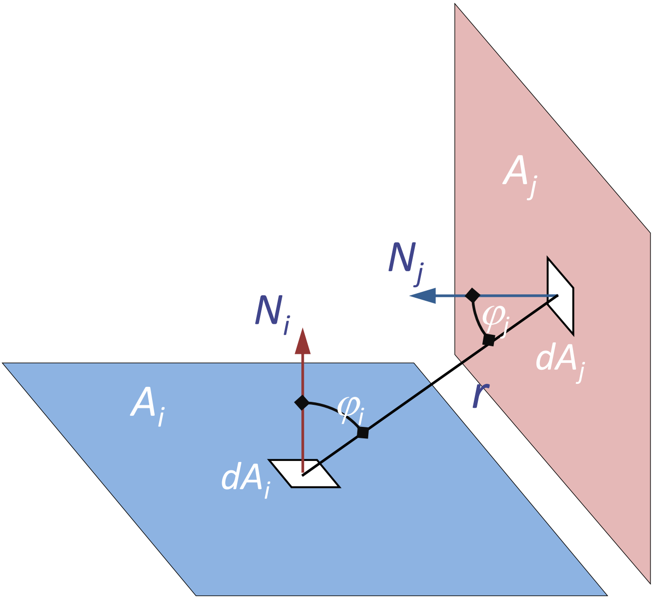
$$B_i = E_i + \rho_i \sum_{j=1}^{n} B_j F_{ij}$$

- Where E_i represents the energy emitted by patch *i*, ρ_i the reflectivity parameter of patch *i*, and $\sum_{j=1}^{n} B_j F_{ij}$ the energy reflected to patch *i* from the *n* patches *j* around it, depending on the *form factors* F_{ij}.

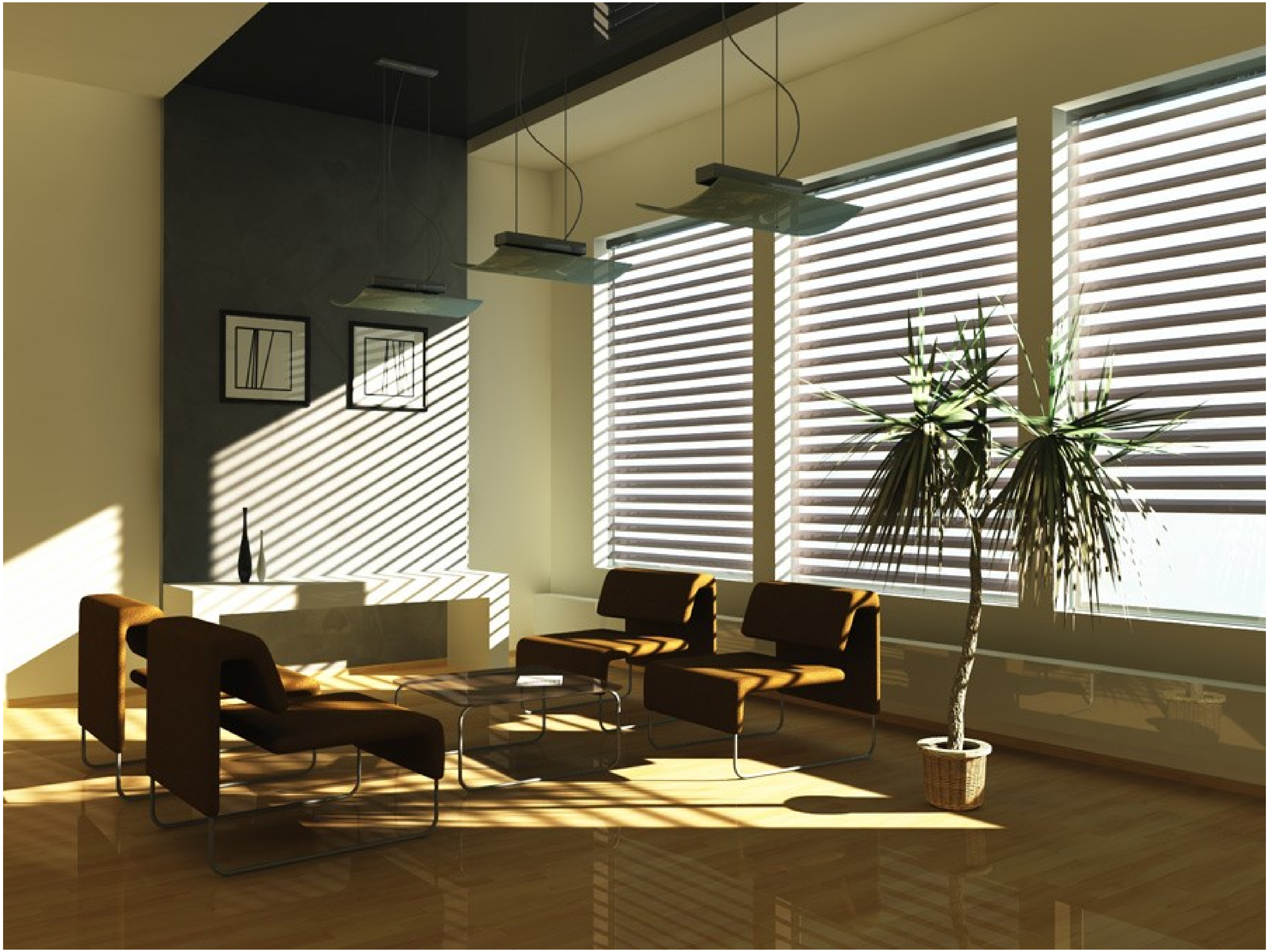
- The form factor represents the fraction of light that reaches patch *i* from patch *j*. It depends on the distance and orientation of the two patches.

- *A scene with n patches, follows a system of n equations for which the solution yields the radiosity of each patch.*



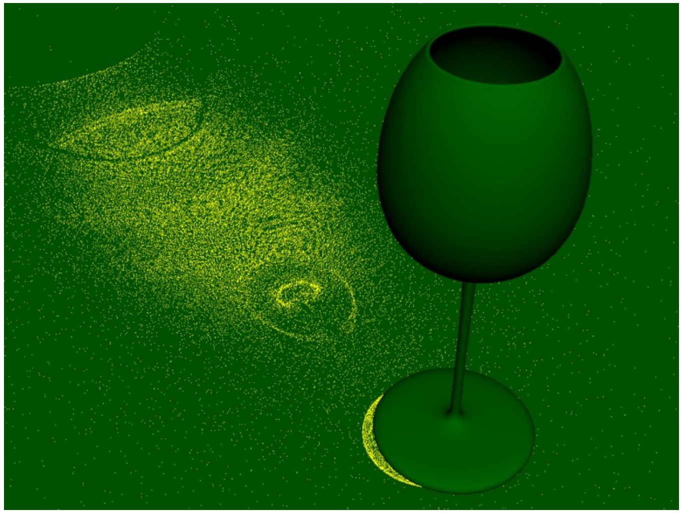


Example

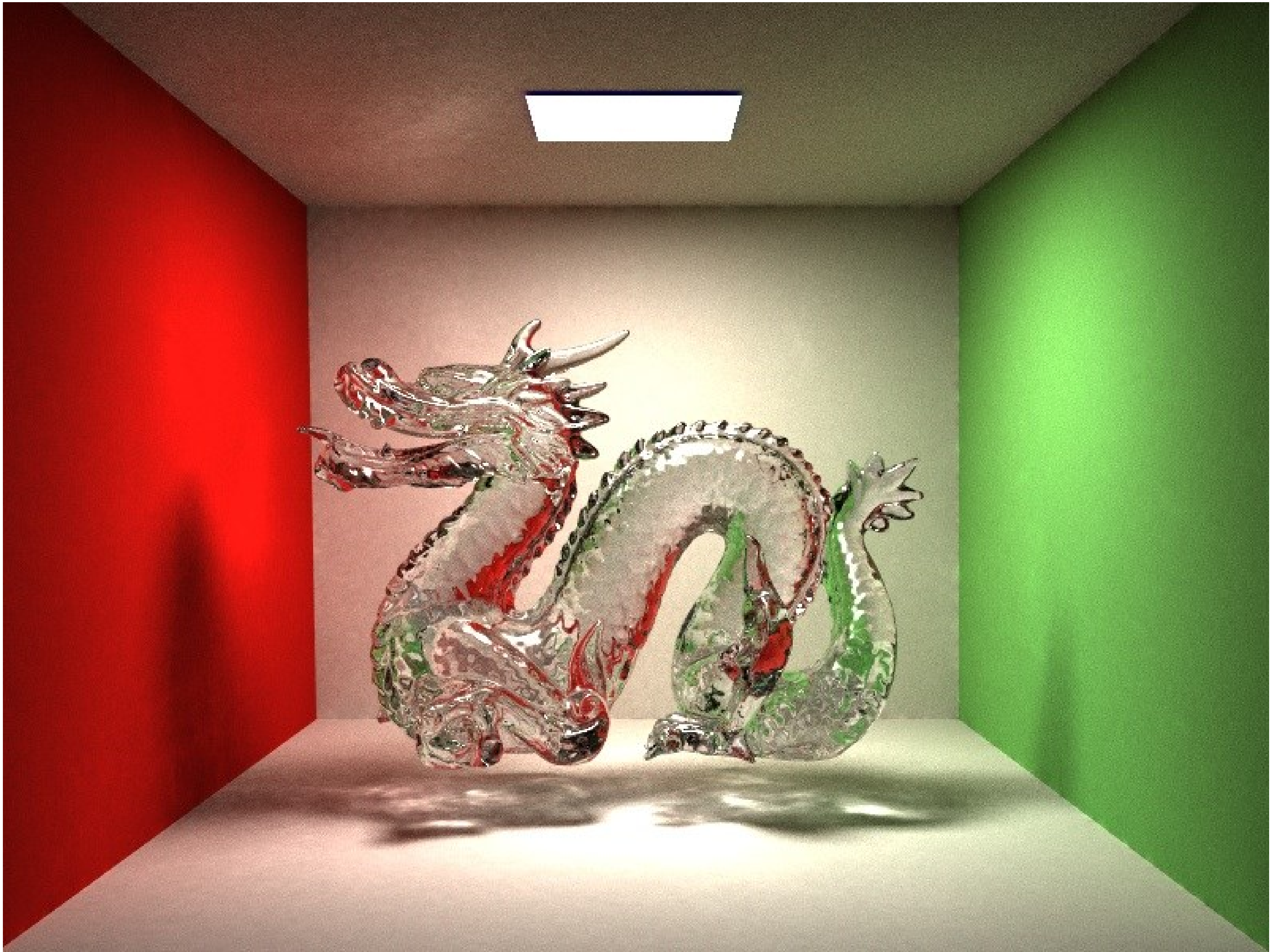


Global illumination photon mapping

- **Photon mapping** is a two-pass algorithm developed by Henrik Wann Jensen that approximately solves the rendering equation.
- Rays from the light source and rays from the camera are traced independently until some termination criterion is met, then they are connected in a second step to produce a radiance value.
- It is used to realistically simulate the interaction of light with different objects. Specifically, it is capable of simulating the refraction of light through a transparent substance such as glass or water, diffuse interreflection between illuminated objects, the subsurface scattering of light in translucent materials, and some of the effects caused by particulate matter such as smoke or water vapor.

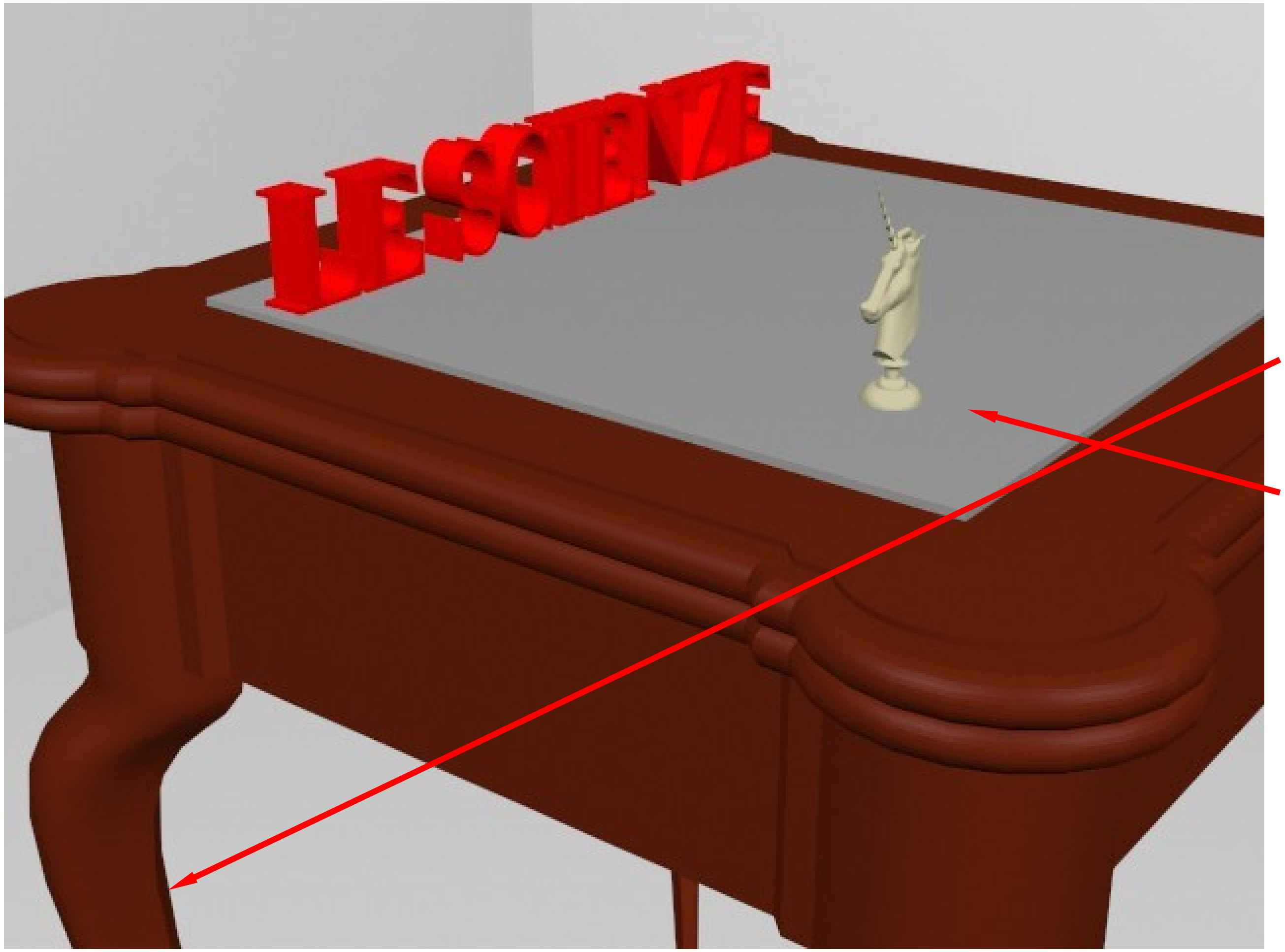


Example



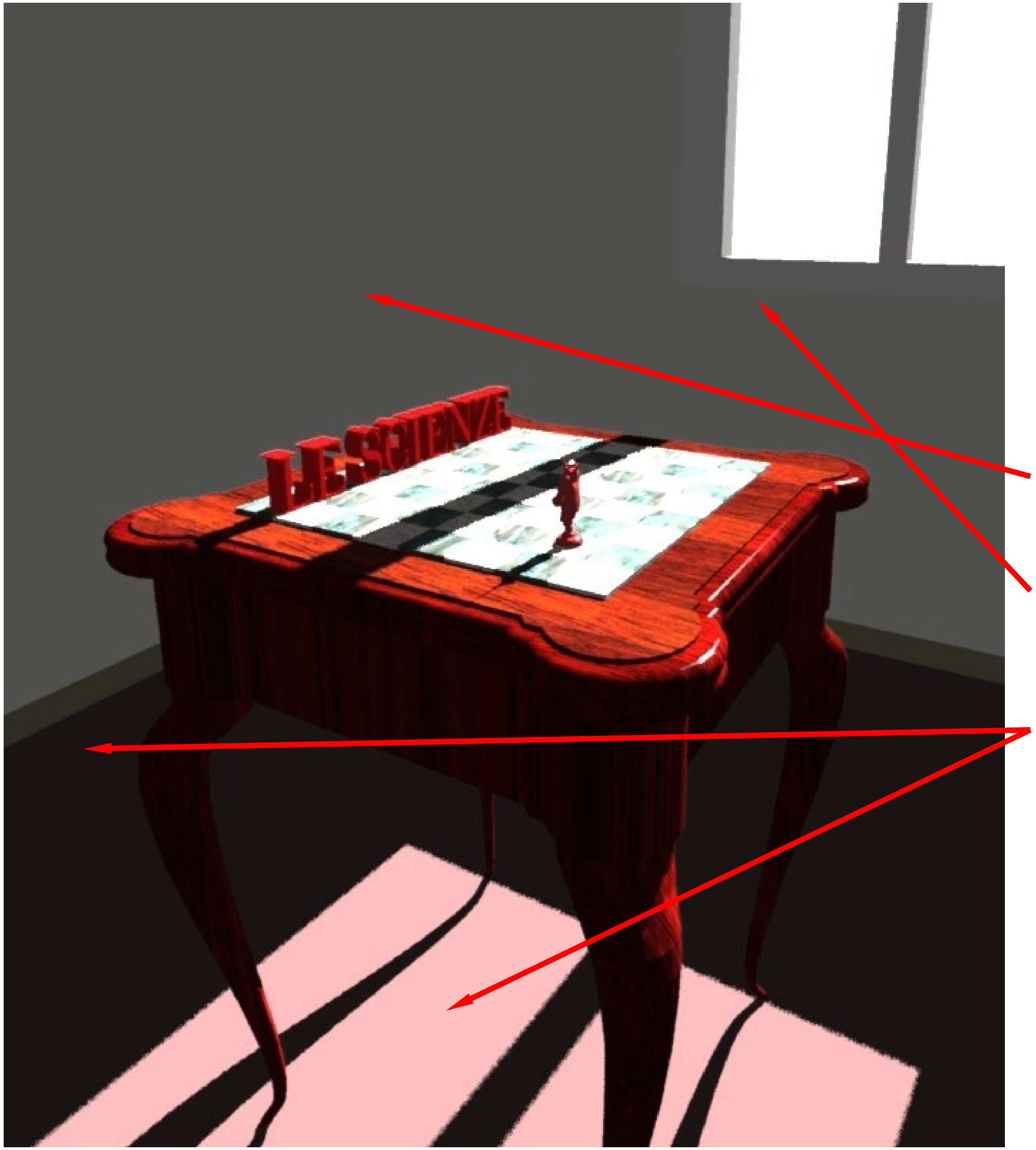
A comparison by Ledah Casburn

…. Phong model



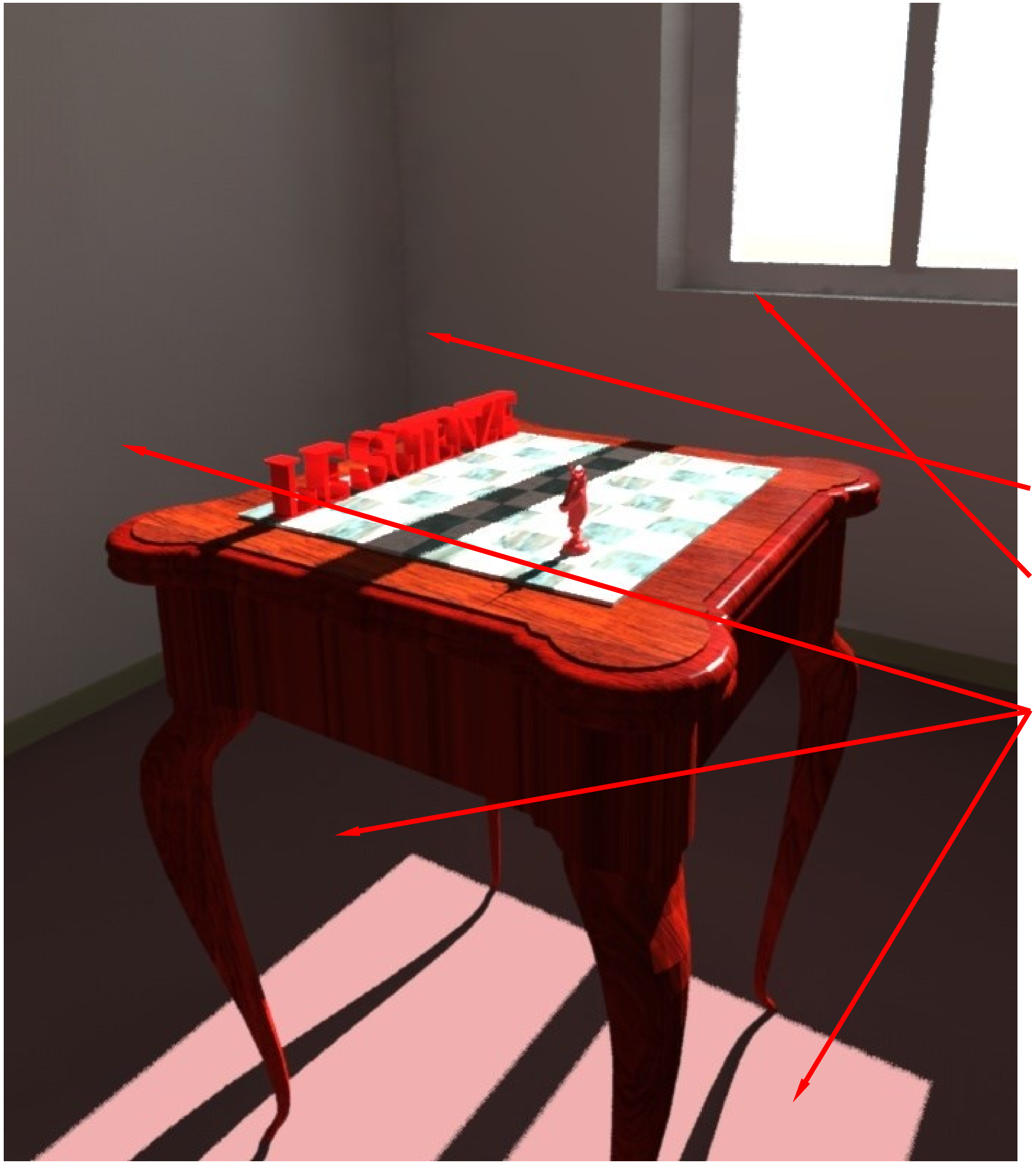
- No objects interactions
- No shadows

… ray tracing



- is there a corner?
- is the window in depth?
- is the floor properly rendered?

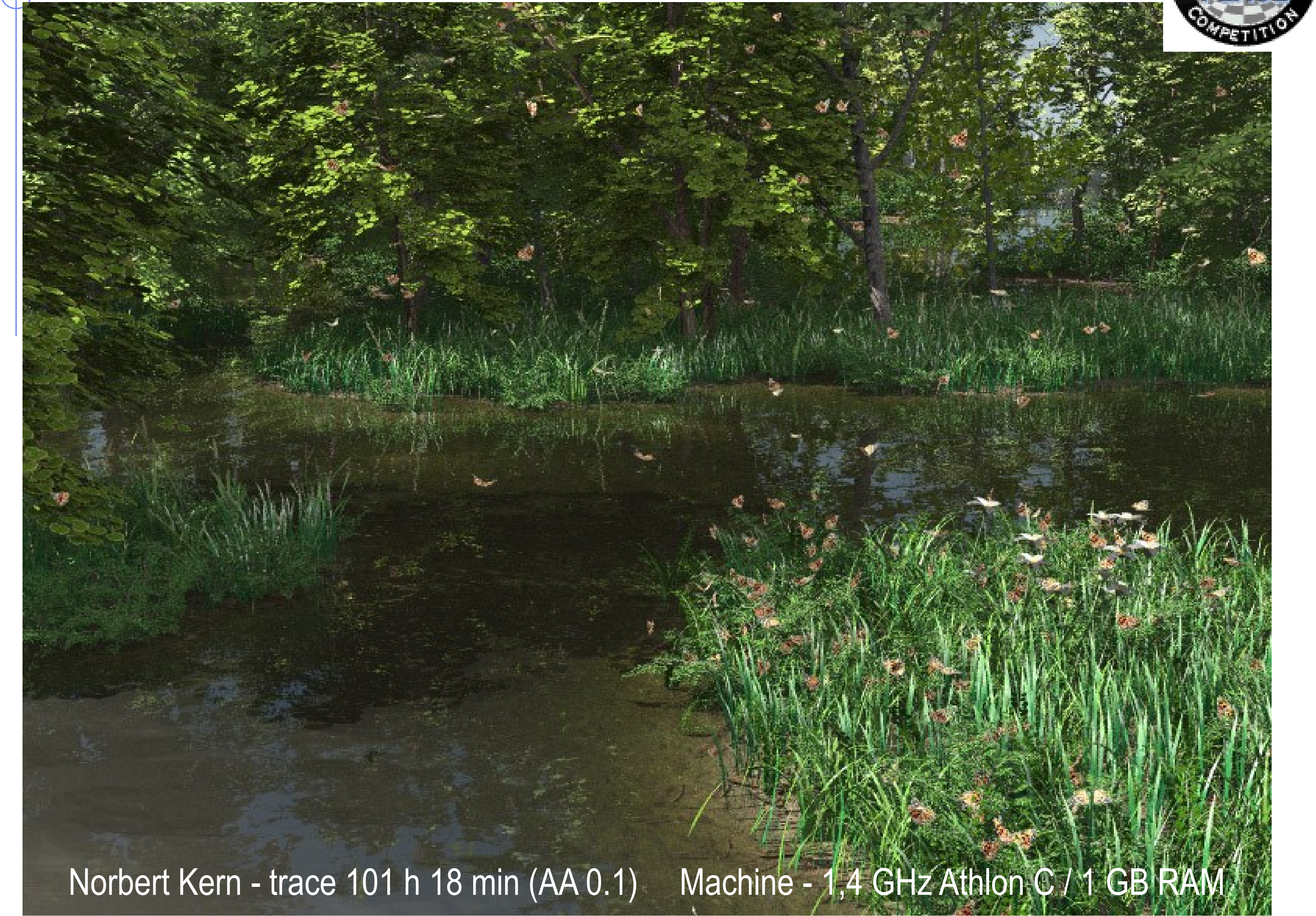
.. radiosity



- there is a corner
- the window is in depth
- floor and walls are clearer and pink

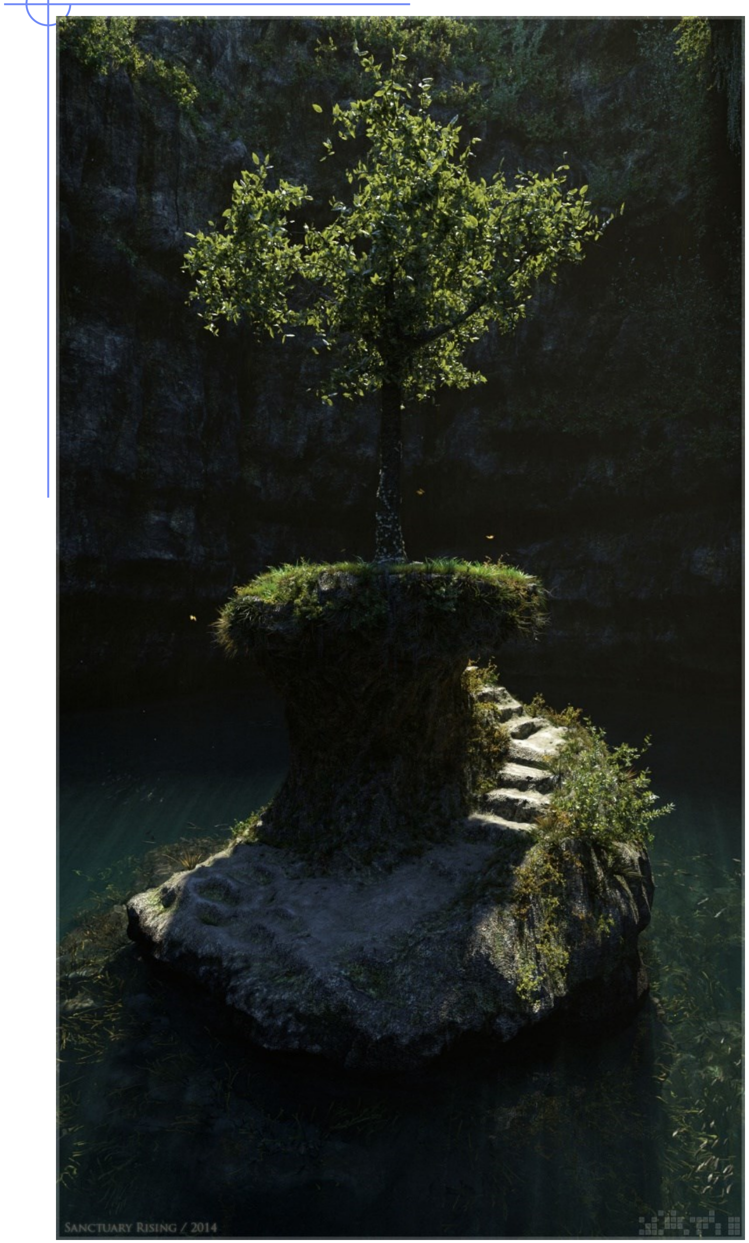
Award winning examples…
July-August 2000

INTERNET RAYTRACING 2nd PLACE COMPETITION



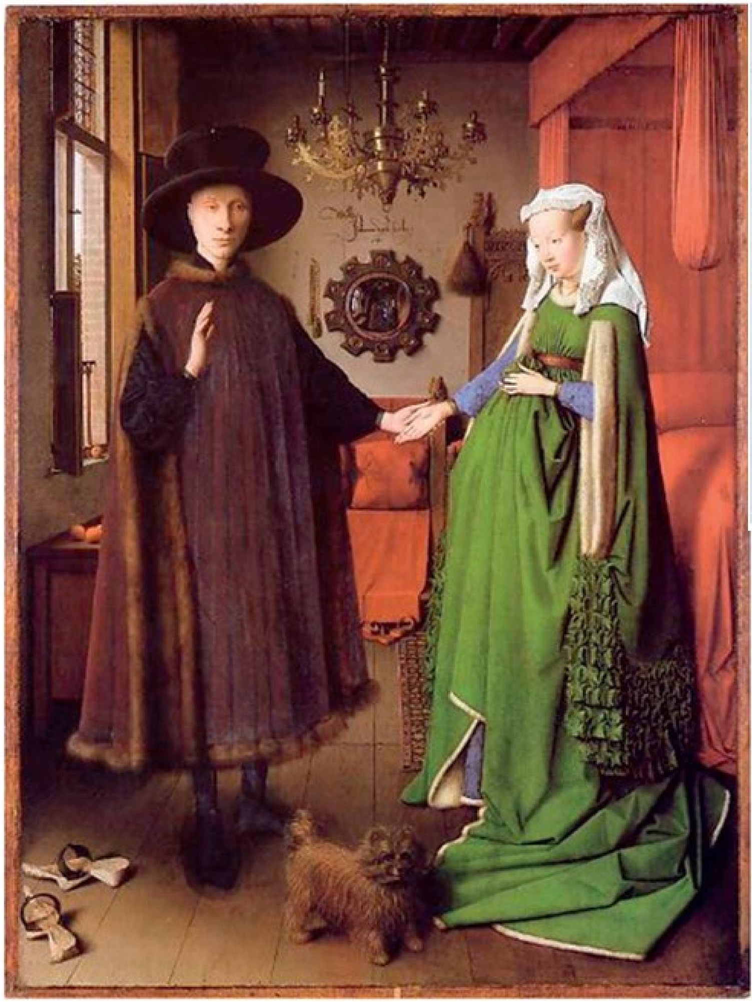
Norbert Kern - trace 101 h 18 min (AA 0.1)   Machine - 1,4 GHz Athlon C / 1 GB RAM

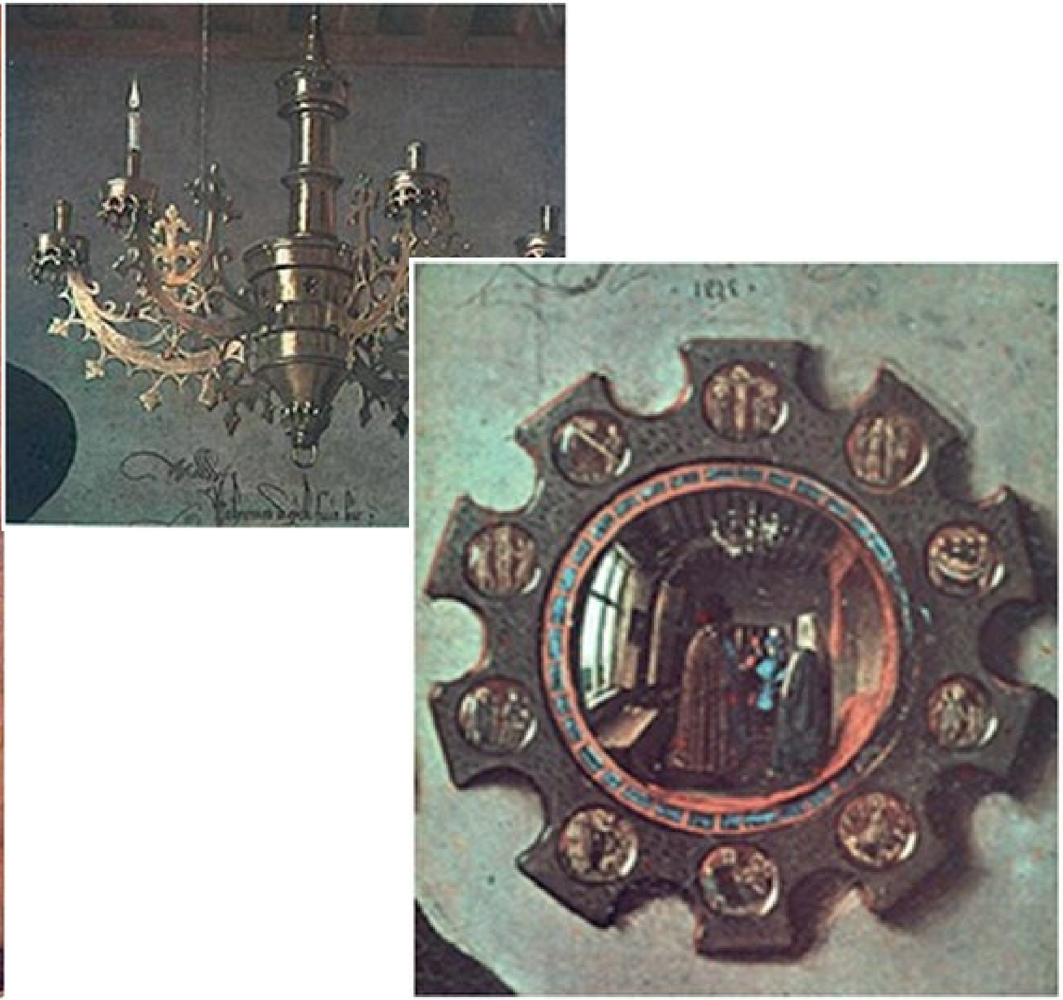
Example: Render of the year 2014



The betrothal of the Arnolfini

Jan Van Eyck 1434, National Gallery, London





Examples

Materials and Texture Maps

- Using materials and textures can create attributes such as colour, transparency, and shininess of an object. Materials control how the light interacts with the an object's surface
- Textures are image that can be 'mapped' to geometry to provide authentic surface detail and characteristics.
- Texture maps are useful for providing a look for an object where a high level of detail is not required for an objects geometry - brick walls, guitar fret board for example.
- Texture maps can also be used to replace distant geometry where level of detail is such that it permits the substitution of the geometry with a textured primitive object.
- Texture maps can themselves be combined with to provide layered texturing characteristics to achieve the desired modelling effect

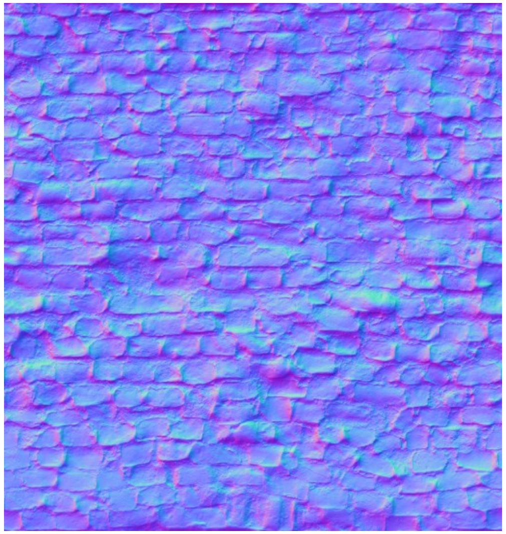
Texture mapping

- Texture mapping maps image onto surface, two approaches:
    - Forward Mapping
        - Each points in the image is mapped
        - Harder to implement correctly, and harder to model
    - Inverse Mapping
        - Each vertex is associated with a point on an image (u,v)
        - Inverse mapping is much more commonly used
        - Most 3D modelers output U,V co-ordinates for texture application

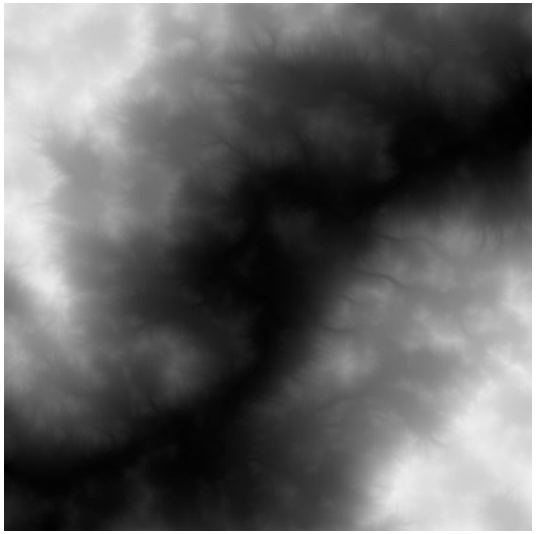
Normal and Height maps

- *Normal map (o bump-map)*: normal maps are a type of texture map that can help give a more realistic indication of bumps or depressions on the surface of a model. Consider a brick wall: to increase the impression of realism, a bump or normal map would be added to more accurately recreate the coarse, grainy surface of bricks, and heighten the illusion that the cracks between bricks are actually receding in space.

Esempio di *Normal map* di mattoni



Esempio di *Height map* di mattoni

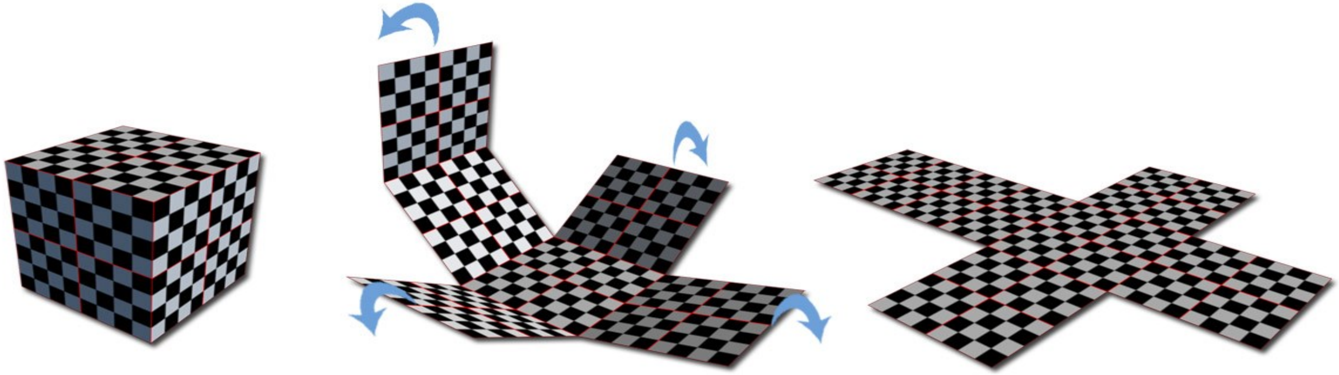


- *Height map*: are typically used to deform terrain meshes moving vertices up and down. Displacement maps are similar to bump but store height information and modify geometry when rendered, modifying both appearance of shading and silhouette.

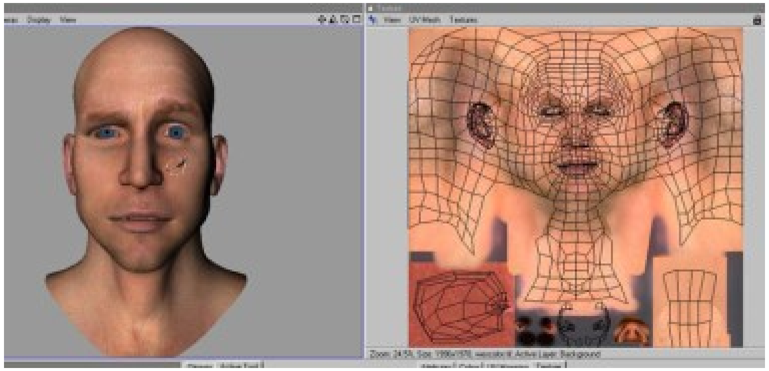
Texture mapping

- UV Mapping is the process of mapping a 2D coordinate grid to a 3D model. The 2D grid is associated with a corresponding image file to place on the model. UV coordinates (also known as texture coordinates) can be generated for each vertex in the mesh. One way is to unfold the mesh at the seams, automatically laying out on a flat page.
- Once the model is unwrapped the scene may be rendered combining each mesh element to the appropriate texture image

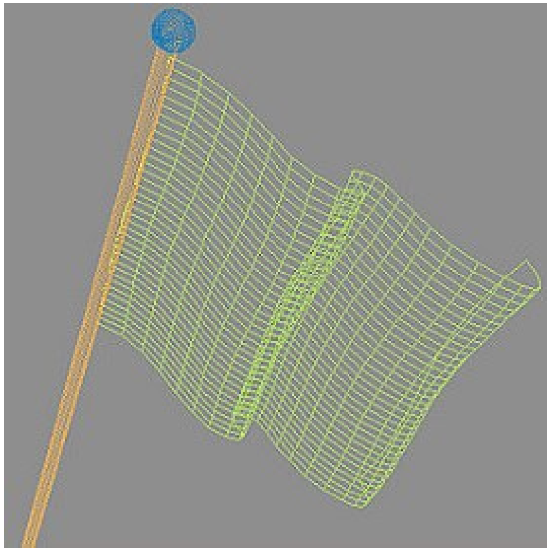
Unwrapping example of cube



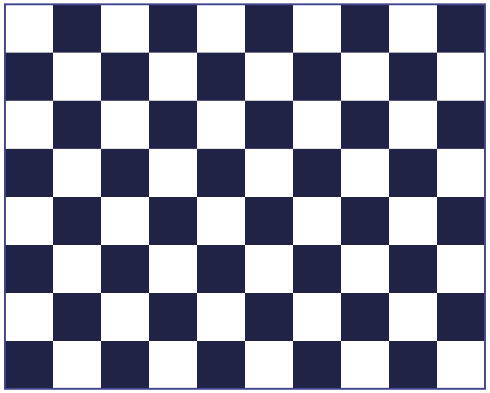
Unwrapping example of head



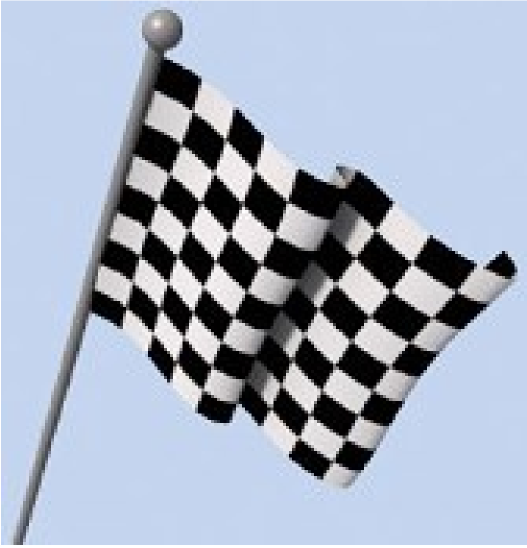
Texture mapping, an objective:



+

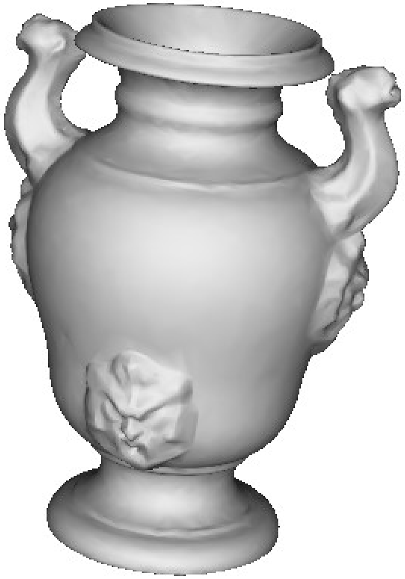


=

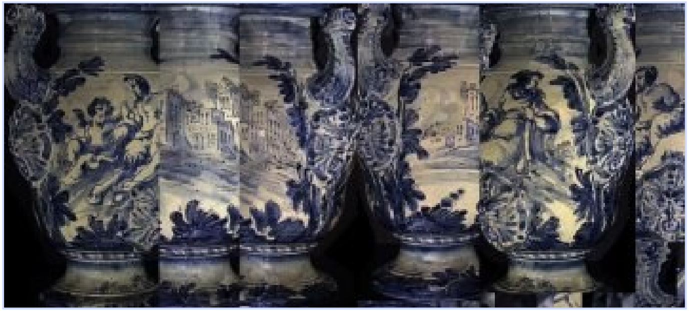


3D mesh

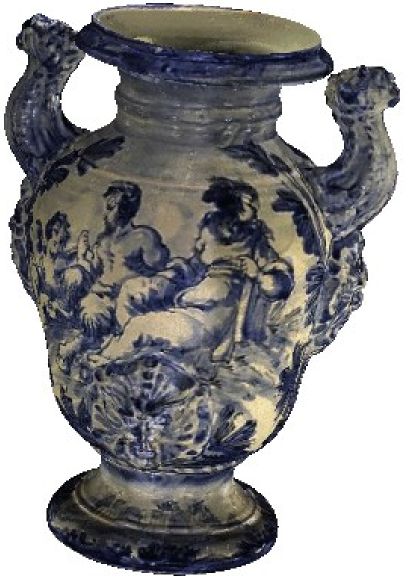
2D texture
(color-map)



+



=

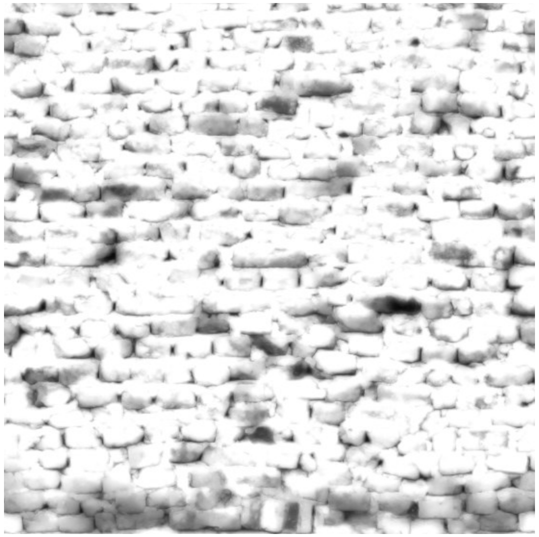


Color and Ambient Occlusion maps

- Texture maps are developed to directly correspond to the UV coordinates of an unwrapped 3D model:
- *Color map*: As the name would imply, the most obvious use for a texture map is to add color or texture to the surface of a model. However, surface maps play a huge role in computer graphics beyond just color and texture. In a production setting, a character or environment's color map is usually just one of the maps that will be used for almost every single 3D model.

Color map example of bricks

AO map example of bricks



- *AO (Ambient Occlusion) map*: are pre-computed calculations of ambient light bounce on a surface, keeping dark shadows to crevices and sharp corners only.

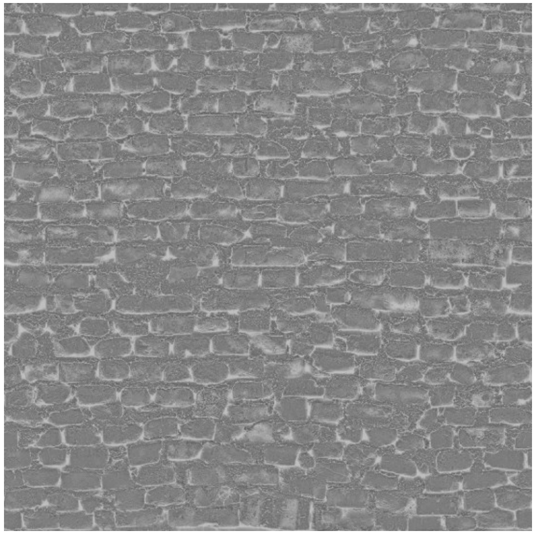
Specular and Roughness maps

- *Specular map (or shininess or gloss map)*: A specular map tells the software which parts of a model should be shiny or glossy, and also the magnitude of the glossiness. Specular maps are named for the fact that shiny surfaces, like metals, ceramics, and some plastics show a strong specular highlight (a direct reflection from a strong light source). White areas correspond to the highest shininess meanwhile black areas to the null case.

Specular map example of bricks

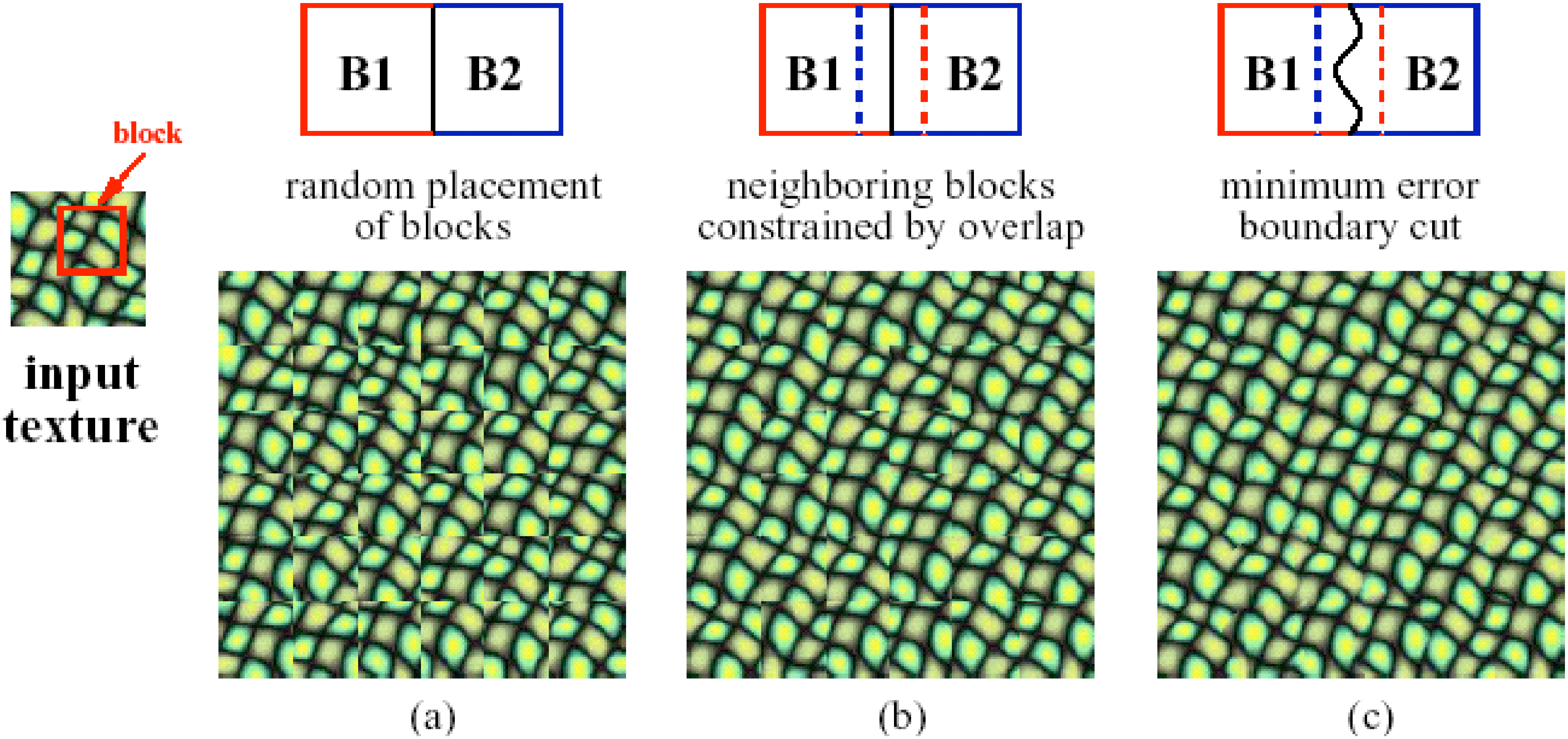
(lack of specularity)

Roughness map example of bricks



- *Roughness/Reflective map*: tells the software which portions of the 3D model should be lambertian. If a model's entire surface is reflective, or if the level of reflectivity is uniform a reflection map is usually omitted. Reflection maps are grayscale images, with black indicating 0% (100%) reflectivity and pure white indicating a 100% (0%) reflective surface. *Environment maps* are typically used for reflective surfaces, showing the scene around the model, what is being reflected. This is a cheap hack to avoid rendering an accurate view of the actual scene, which is typically too slow to render in real time.

Seamless problem



random placement of blocks

neighboring blocks constrained by overlap

minimum error boundary cut

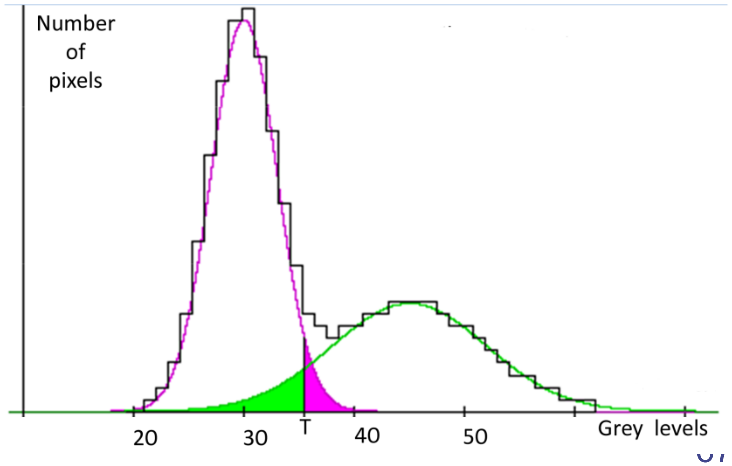
(a)

(b)

(c)

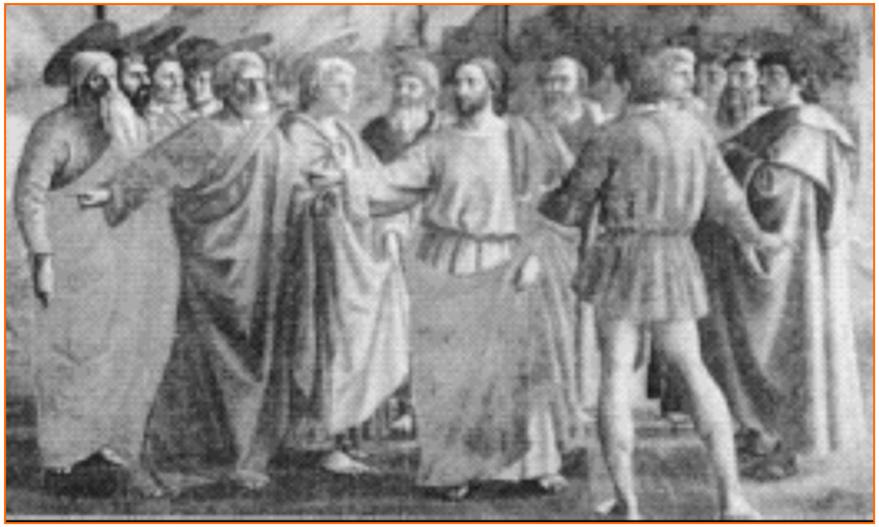
Intensity re-distribution and rendering

- An image histogram has in the abscissa the pixel grey (or color) value while the ordinate plots the number of occurrences of such value.
- Such histogram may be manipulated so as: to obtain a sharper image, to highlight some chromatic distributions, to correct over or under exposed images, to optimize the contrast distribution, etc..
- The information conveyed by histogram is limited to the first order statistics, corresponding to the number of pixels having a given grey value, or color component. Working on the histogram, gross level repartitions can be extracted such as dark vs. bright parts dominant levels (e.g. local maxima) up to splitting the histogram into a conjunction of bounded repartitions.
- In order to distinguish automatically an object from its background a typical approach is:
- i) firstly produce the grey level histogram (this process applies on the RGB components as well);
- ii) find the best bi-Gaussian fit to the histogram distribution;
- iii) compute the threshold T by minimizing the misplaced pixels (either background pixels in the foreground or vice-versa). The misplaced pixel are represented by the green (levels approximately between 25 and T) and magenta (levels between T and 40) values respectively

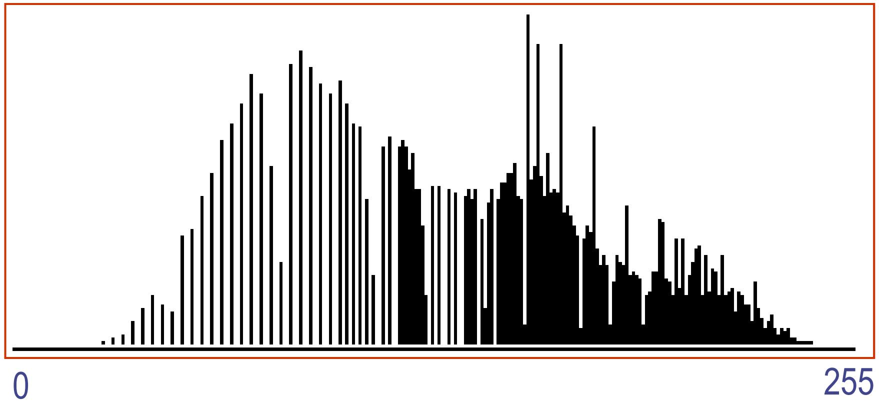


Original image

- Image: History of Saint Peter by Masaccio, around 1427, Church of the Holy Spirit, Florence



- Histogram

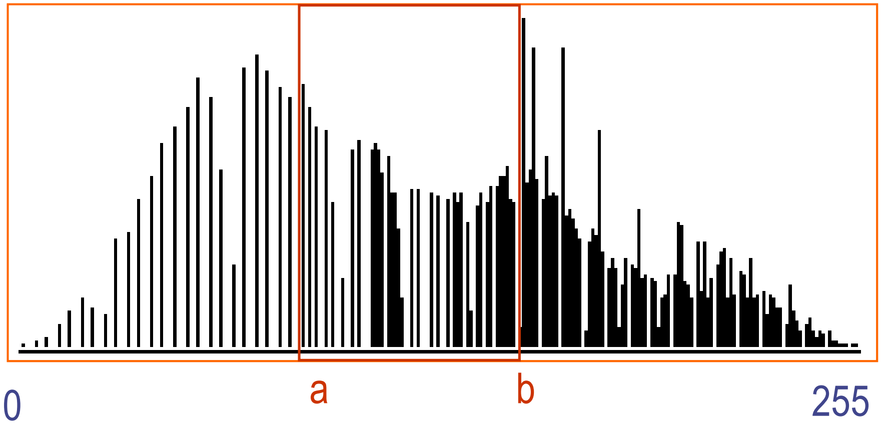


Extended dynamics

- To profit from the full grey level scale, note that very clear and dark pixels are absent, the pixel distribution may extended by stretching with the following formula:

$$g(i,j) = 255 \frac{g(i,j) - g_{\min}}{g_{\max} - g_{\min}}$$

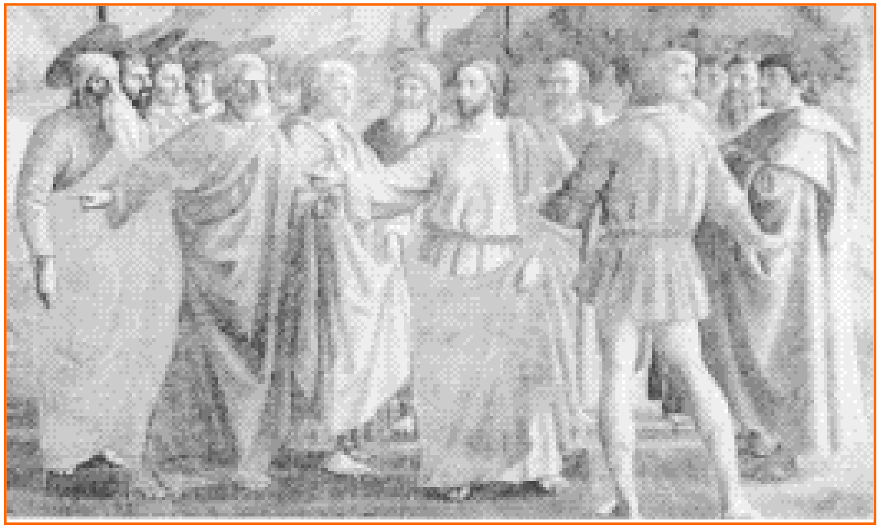
where g_{max}, and g_{min} correspond to the maximum and minimum grey levels respectively, and 255-0 is the whole available grey level scale.

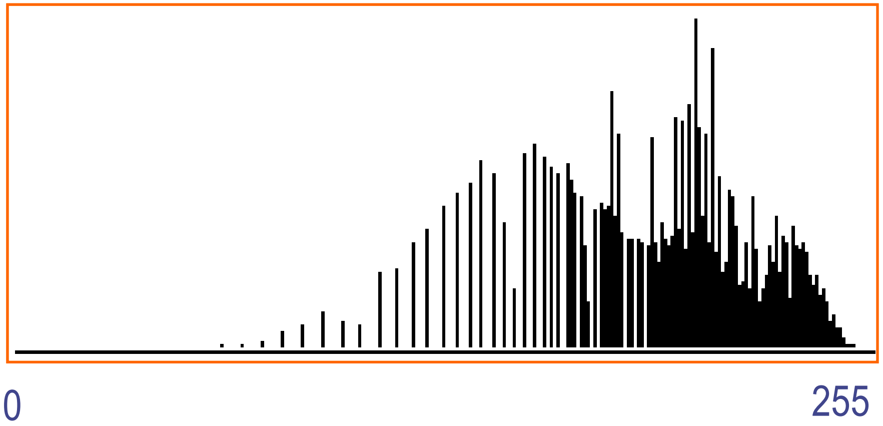


Under-exposure

- To compensate or even create lighting effects as, for instance, under-exposure, a non linear transformation of the grey levels can be performed by:

$$g(i,j)=\sqrt{255g(i,j)}$$

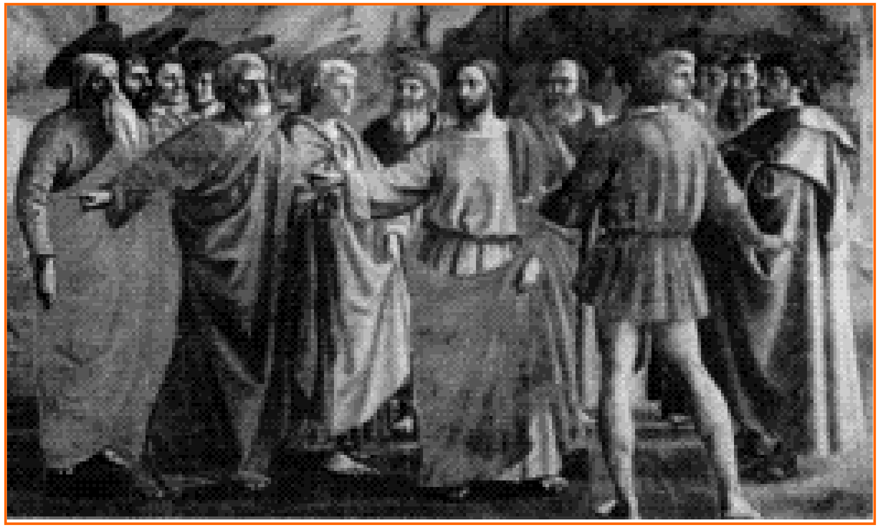


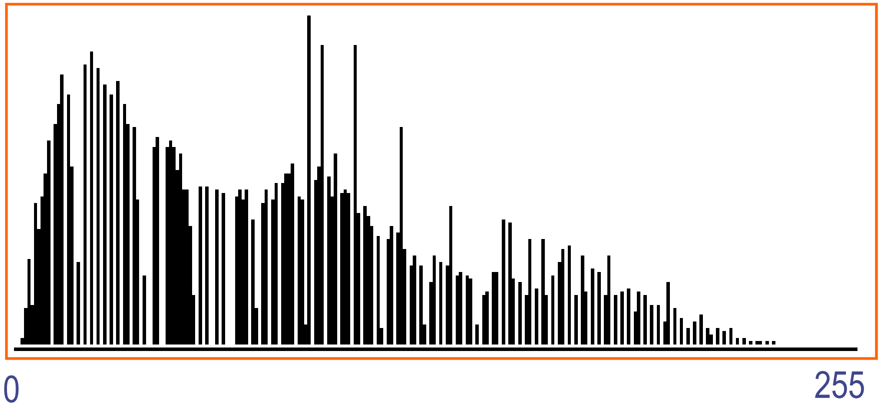


Over-exposure

- To compensate or even create lighting effects as, for instance, over-exposure, a non linear transformation of the grey levels can be performed by:

$$g(i,j) = \frac{g(i,j)^2}{255}$$

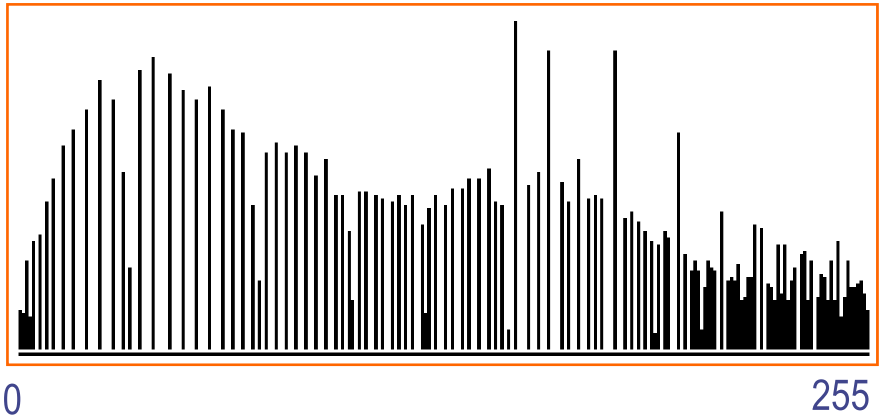




Uniform distribution

- To obtain a *uniform distribution* of image contrast a technique known as *equalization* is employed. This technique consists in making the empiric grey level distribution as close as possible to a uniform distribution, in an adaptive way. The more uniform is a grey level distribution, the better contrasted is the associated image – i.e. *maximal entropy*.

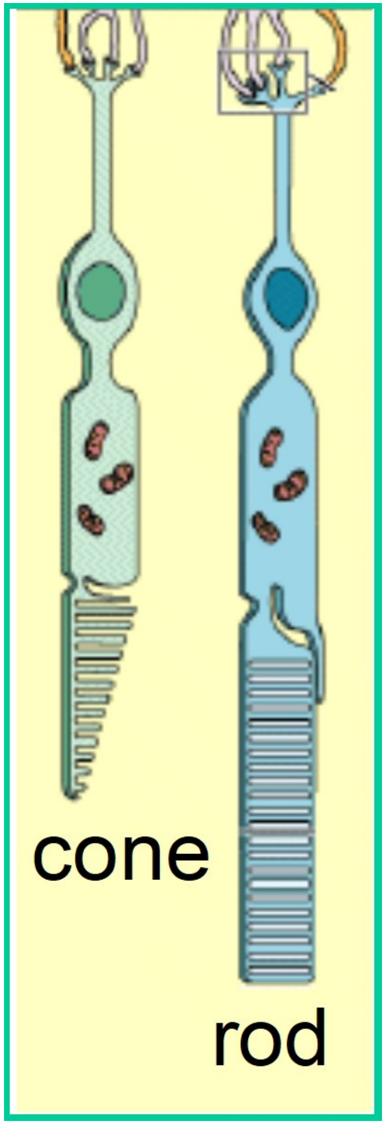
$$g'(i,j) = 255 \frac{\sum_{s=0}^{g(i,j)} h(s)}{\sum_{s=0}^{255} h(s)}$$



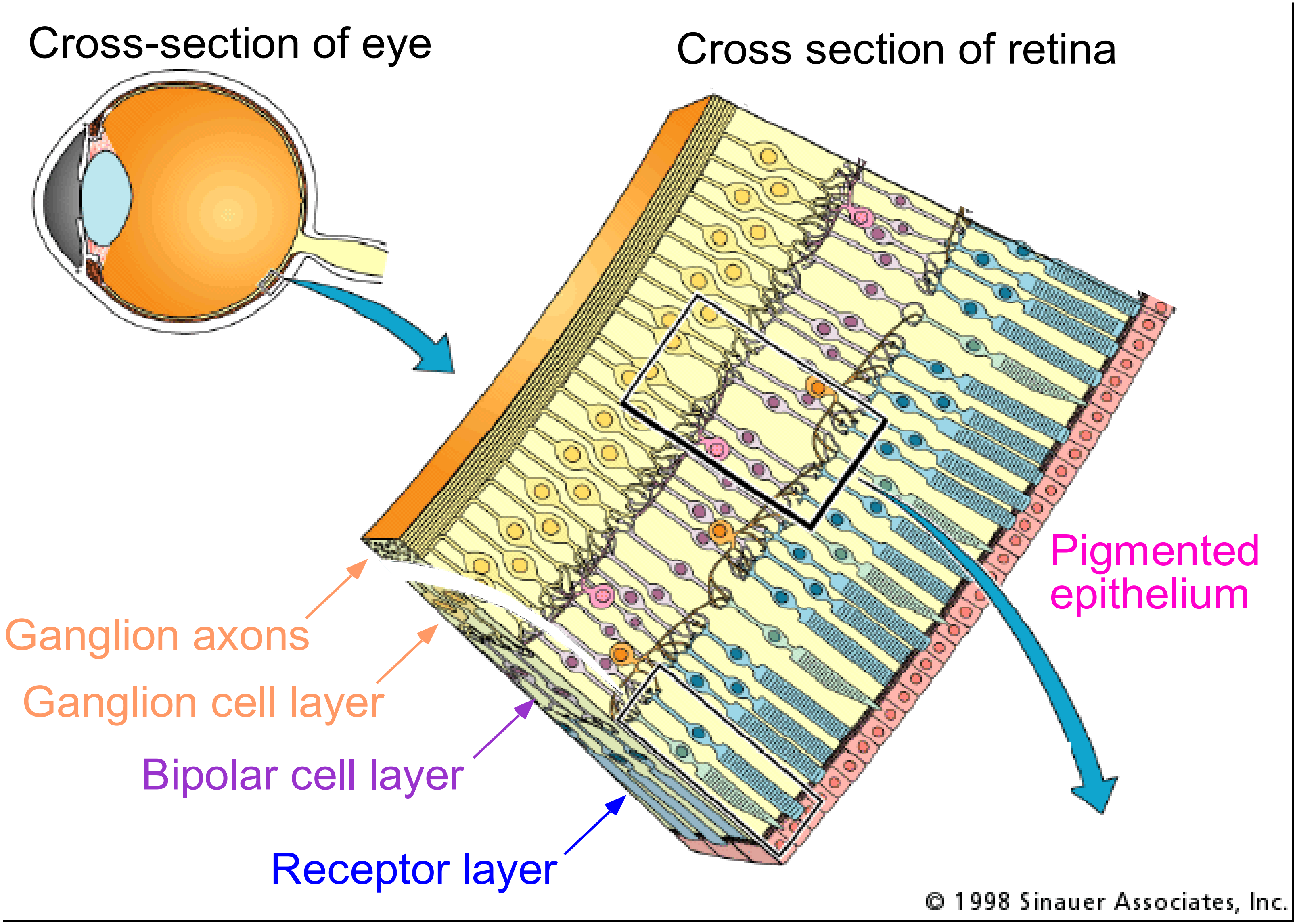
Two types of light-sensitive receptors

Cones
cone-shaped
less sensitive
operate in high light
color vision

Rods
rod-shaped
highly sensitive
operate at night
gray-scale vision



The Retina



Cross-section of eye

Cross section of retina

Ganglion axons

Ganglion cell layer

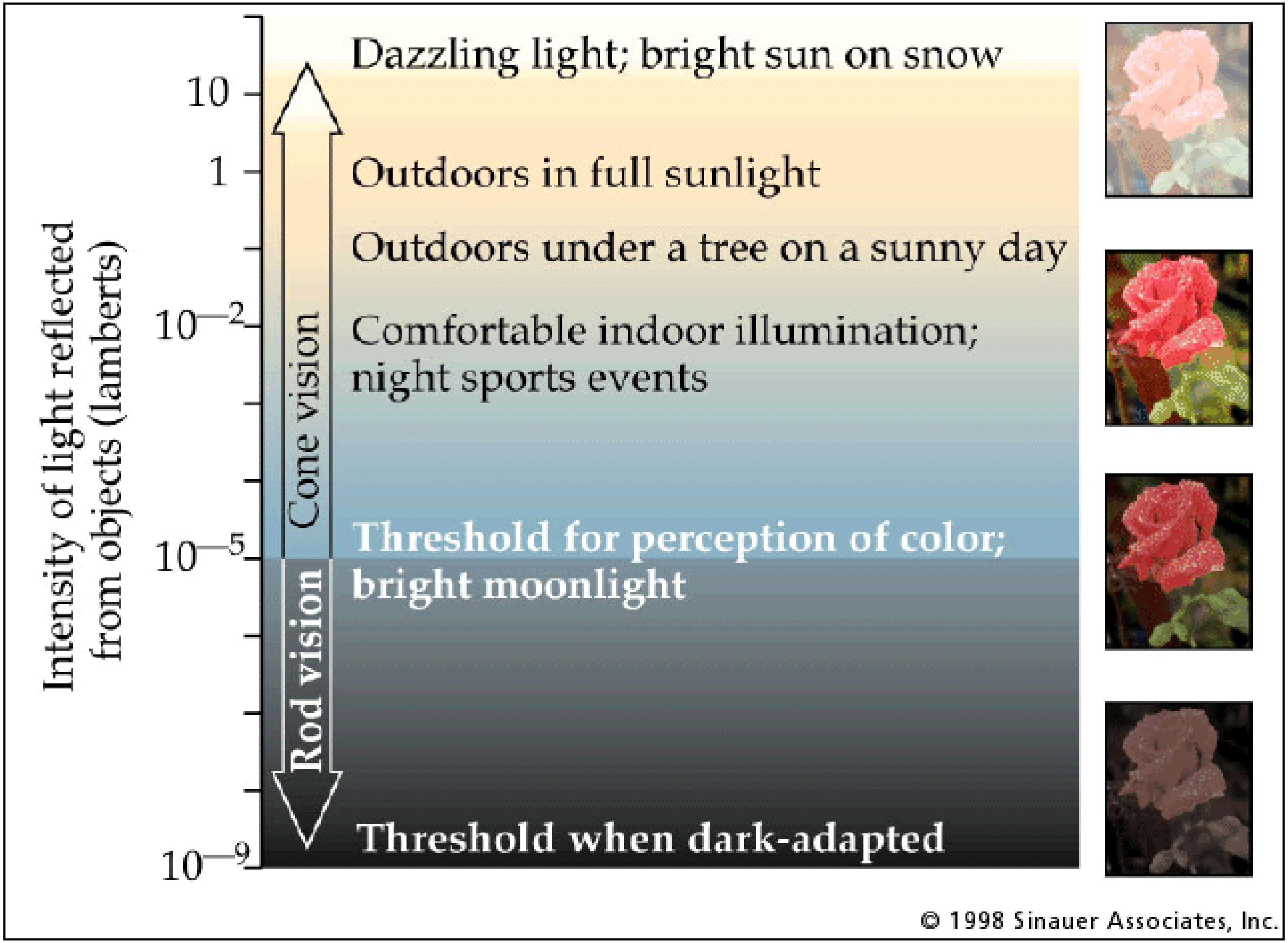
Bipolar cell layer

Receptor layer

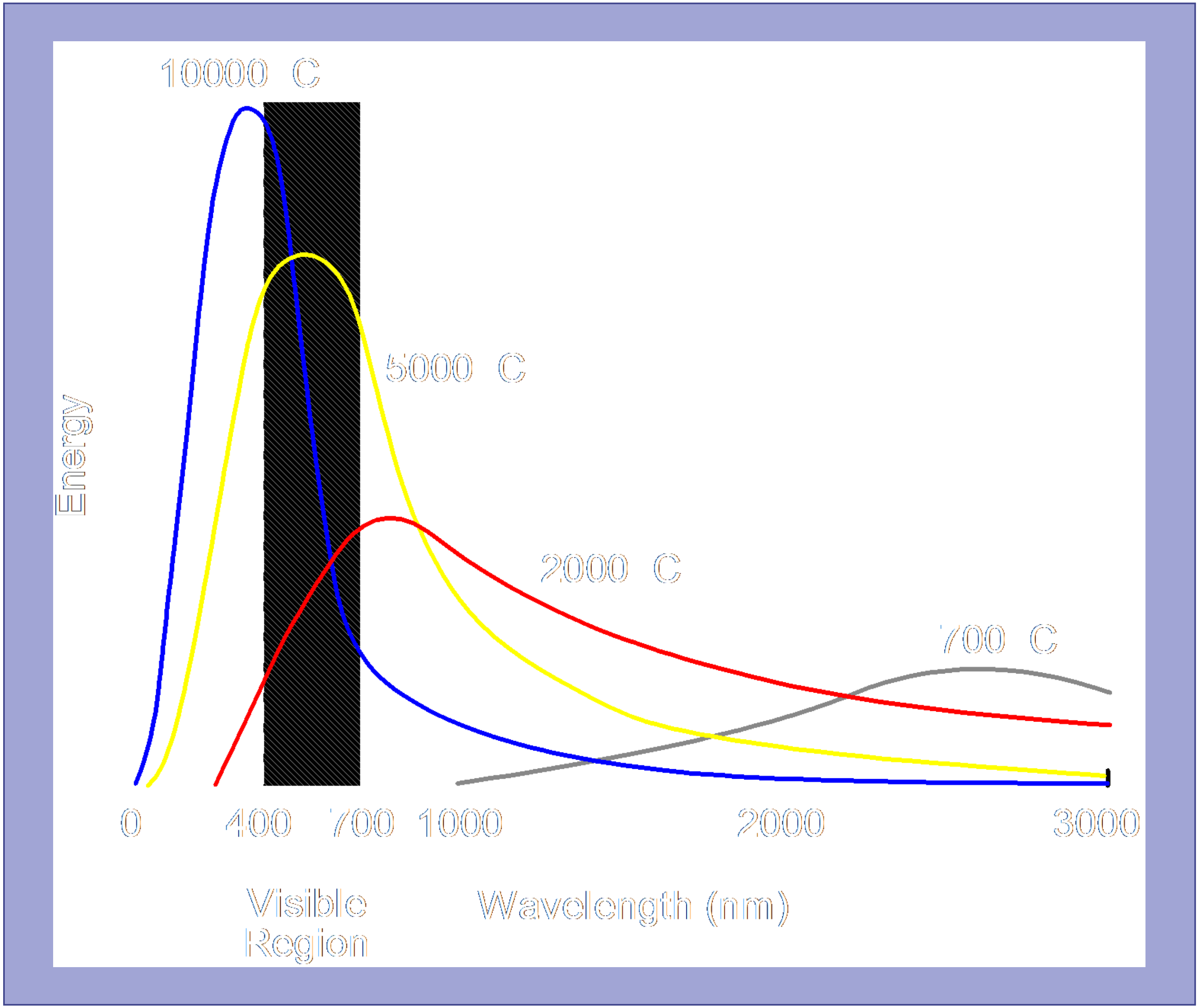
Pigmented epithelium

© 1998 Sinauer Associates, Inc.

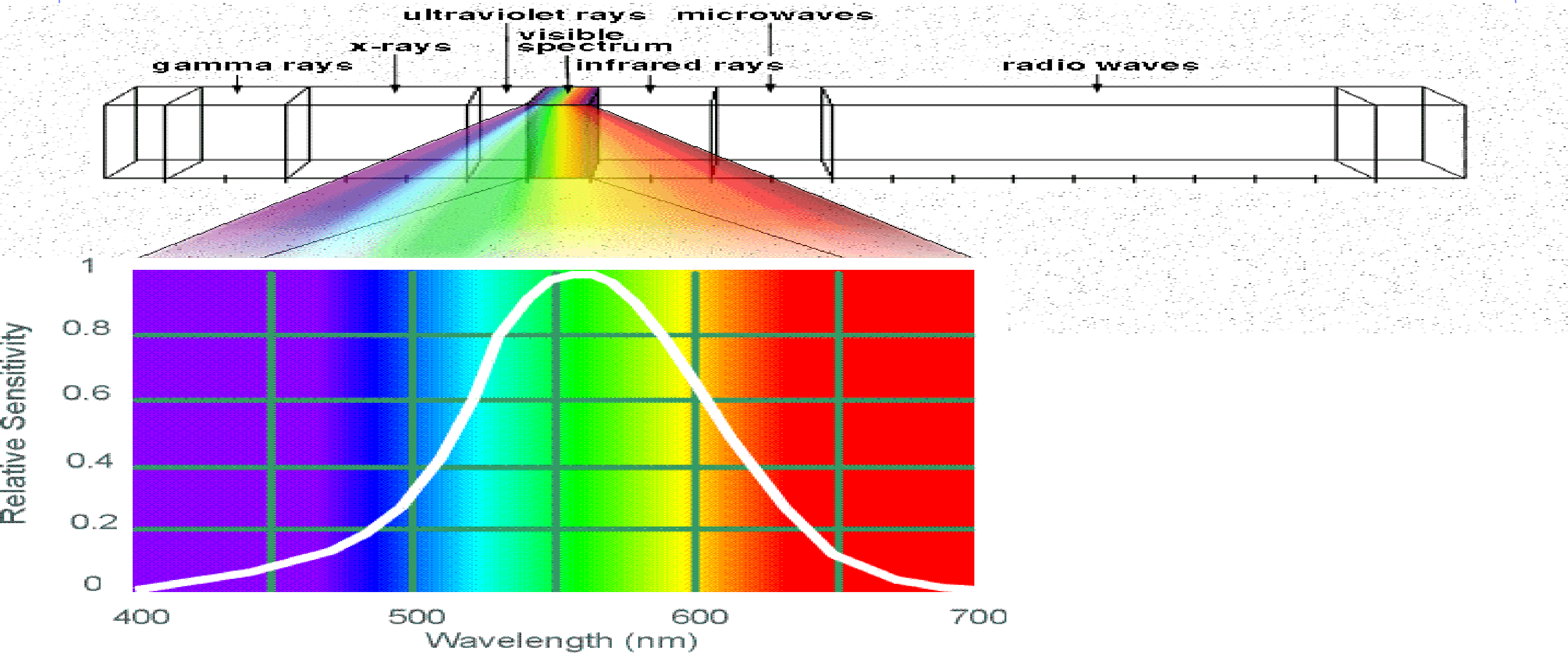
Rod / Cone sensitivity


Intensity of light reflected from objects (lamberts)
10
1
10−2
10−5
10−9
Cone vision
Rod vision
Dazzling light; bright sun on snow
Outdoors in full sunlight
Outdoors under a tree on a sunny day
Comfortable indoor illumination; night sports events
Threshold for perception of color; bright moonlight
Threshold when dark-adapted
© 1998 Sinauer Associates, Inc.

Sun EM energy



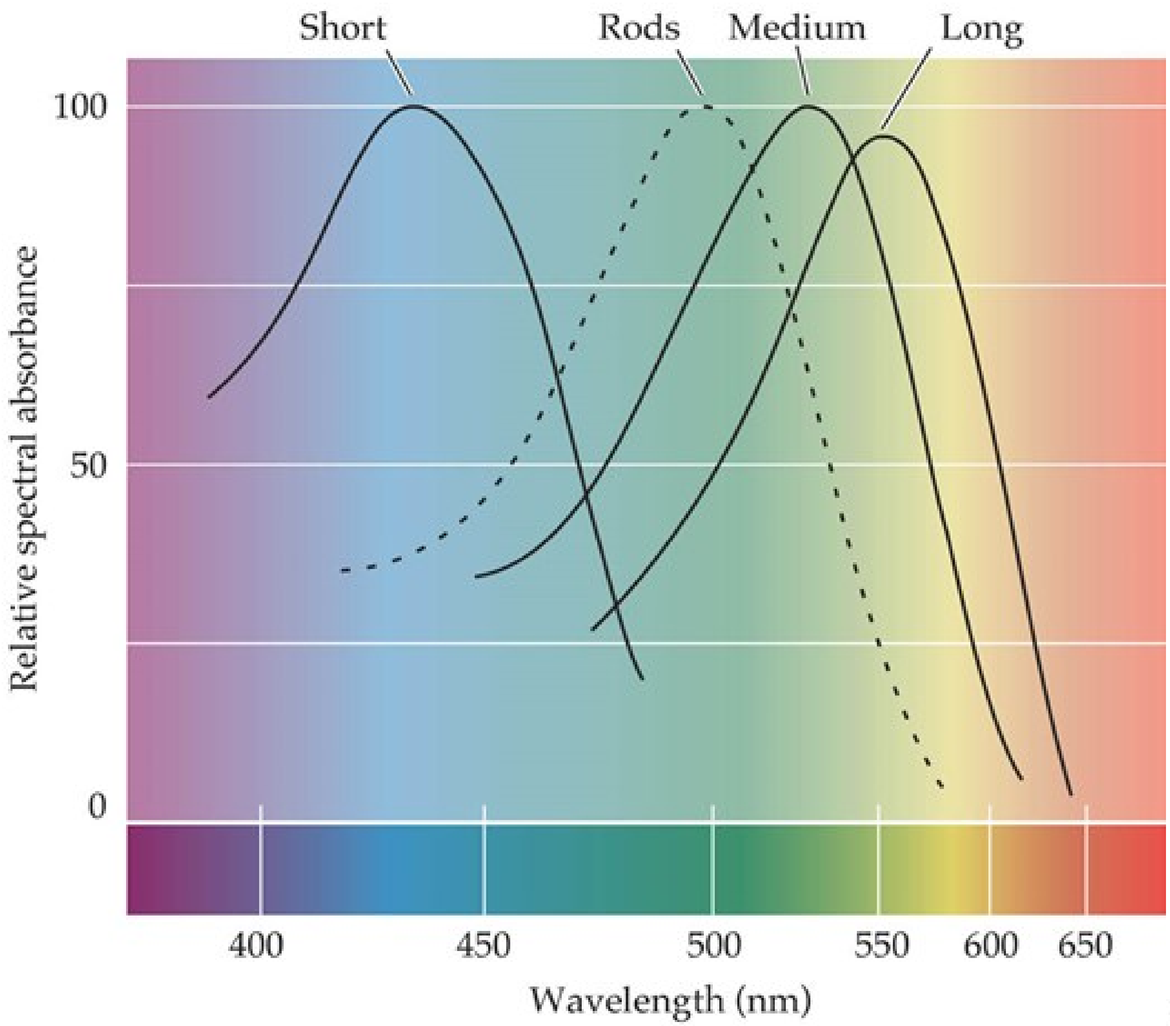
Electromagnetic Spectrum



Human Luminance Sensitivity Function

http://www.yorku.ca/eye/photopik.htm

Human eye color sensors: Types of Cones



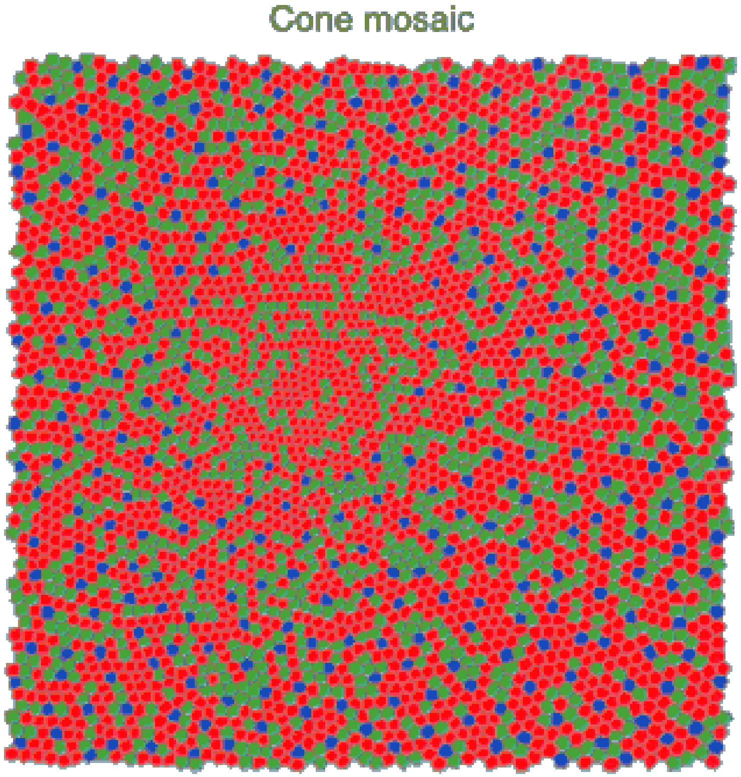
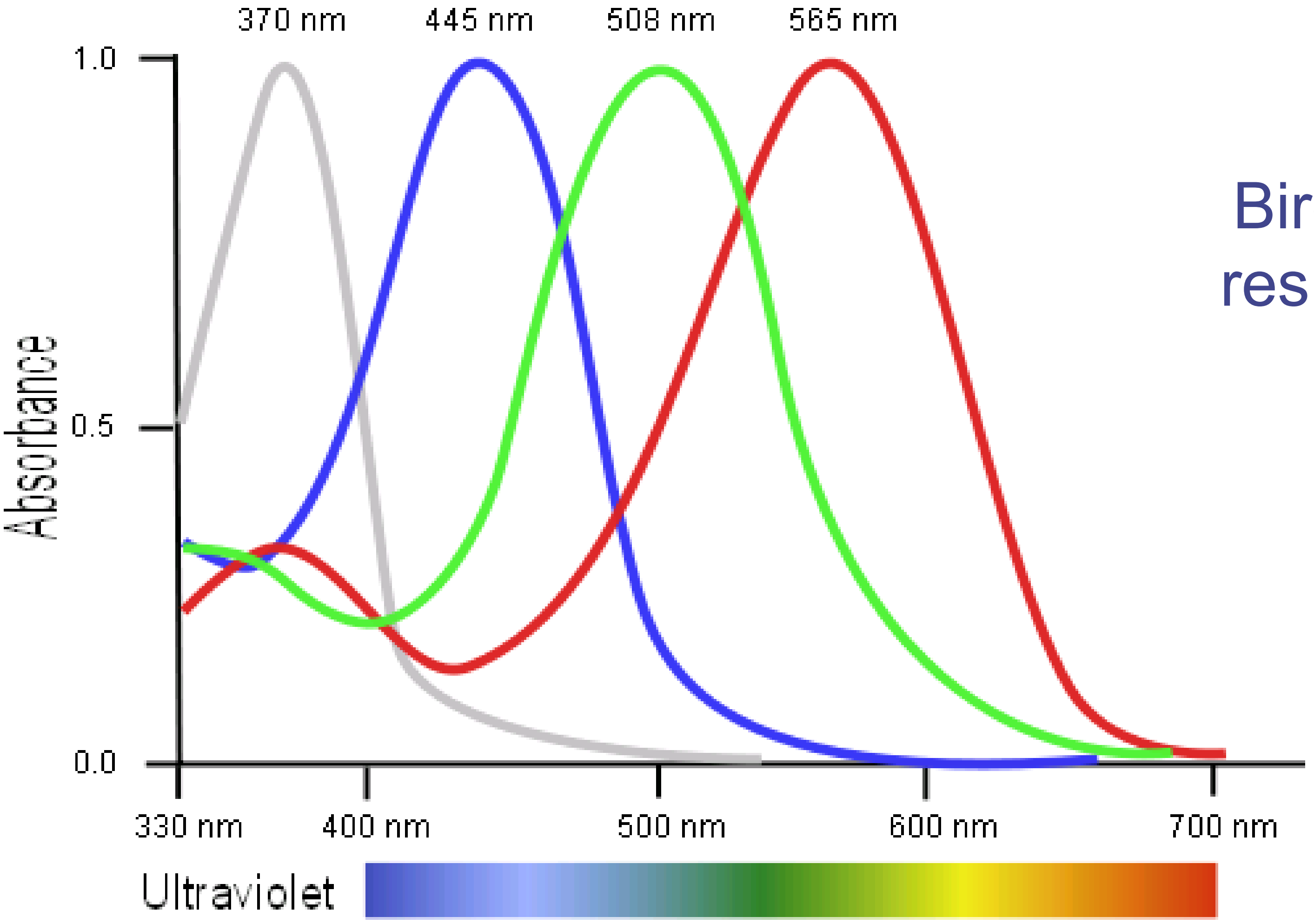


Figure 10.12 © 2004 Sinauer Associates, Inc.

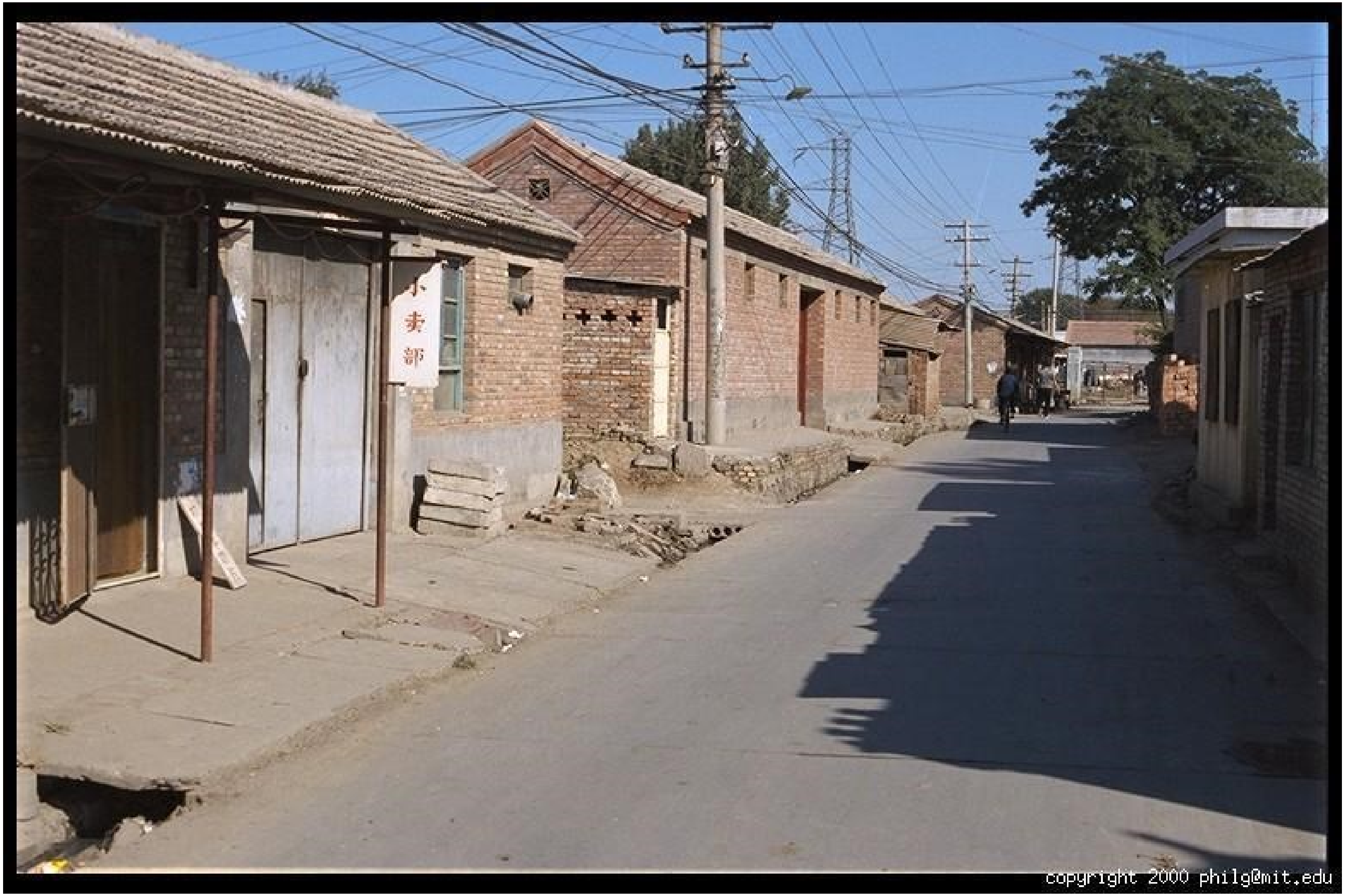
Tetrachromatism

- Most birds, and many other animals, have cones for ultraviolet light.
- Some humans, mostly female, seem to have slight tetrachromatism.



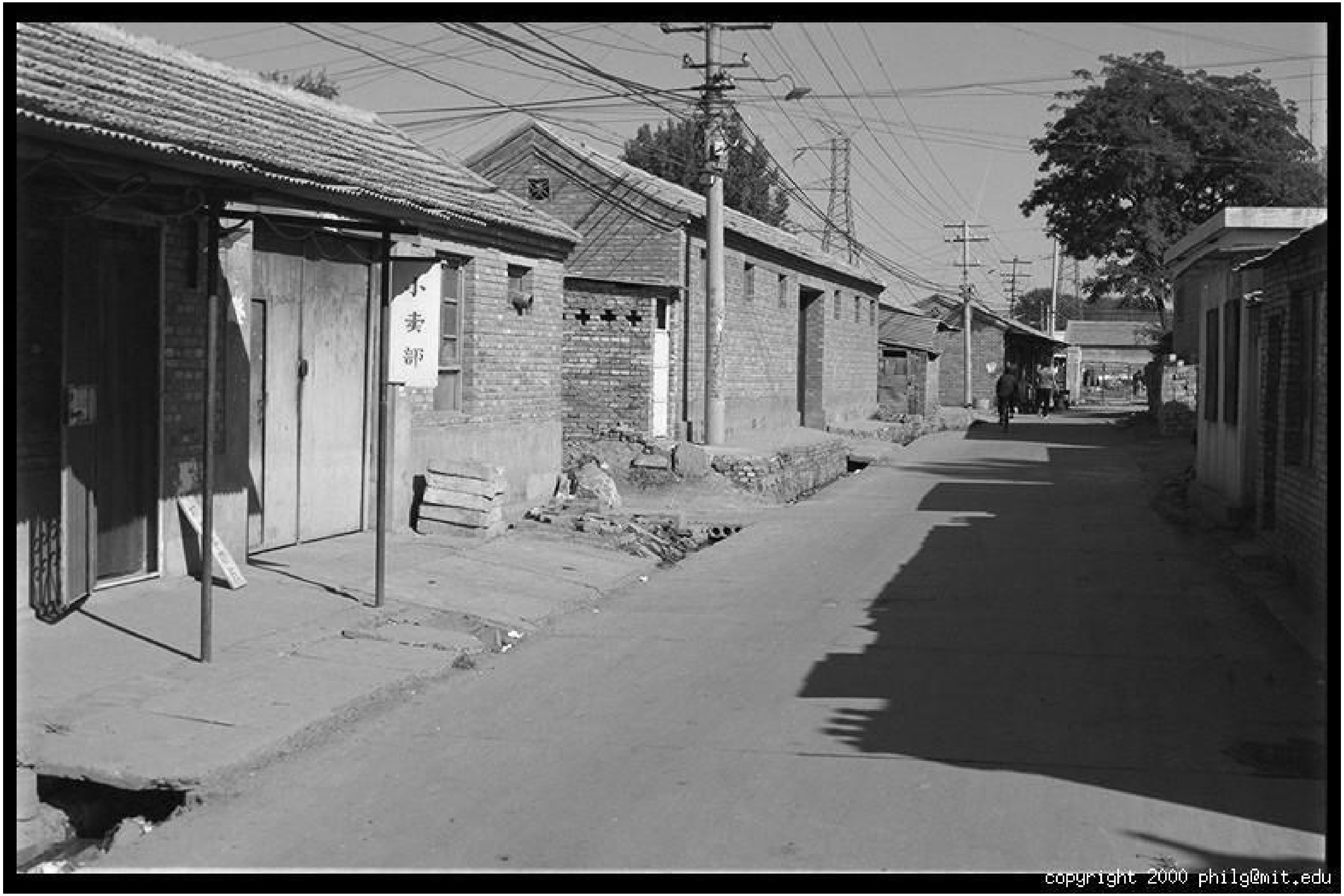
Bird cone responses

Most information in intensity



Original image

Most information in intensity



Only intensity shown – constant color

Standard color spaces

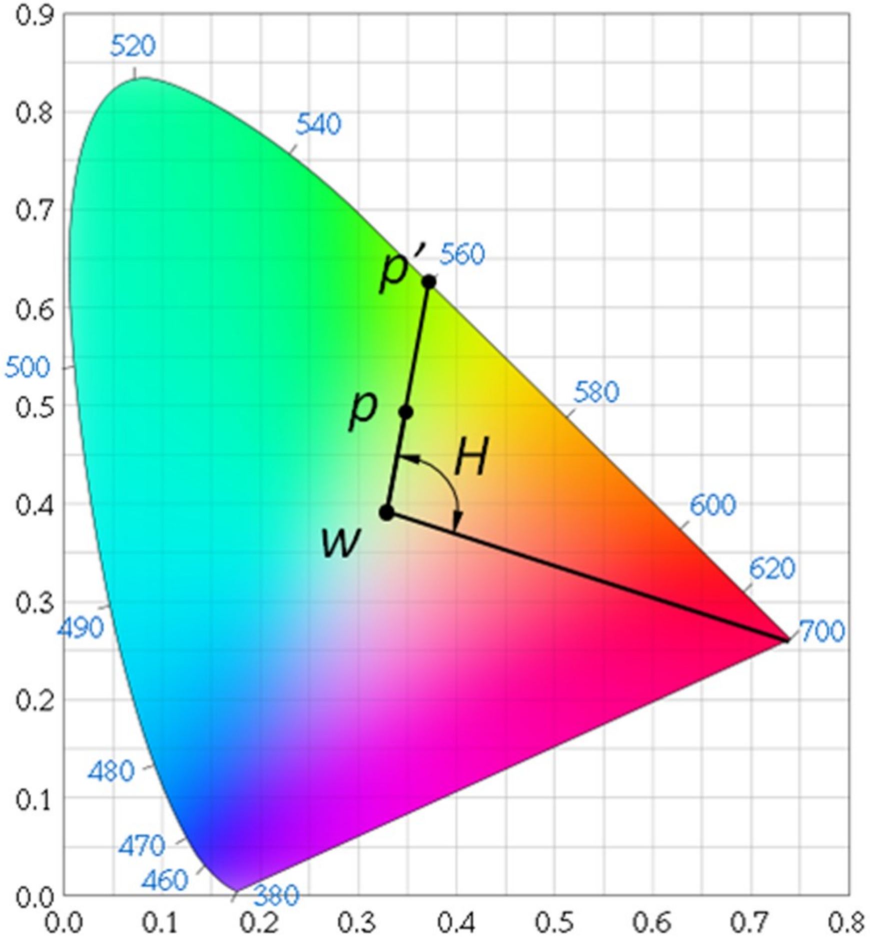
- Use a common set of primaries/color matching functions
- Linear color space examples
  - RGB
  - CIE XYZ
- Non-linear color space
  - HSV

Color representation

$$C = G + B \quad M = B + R \quad Y = R + G$$

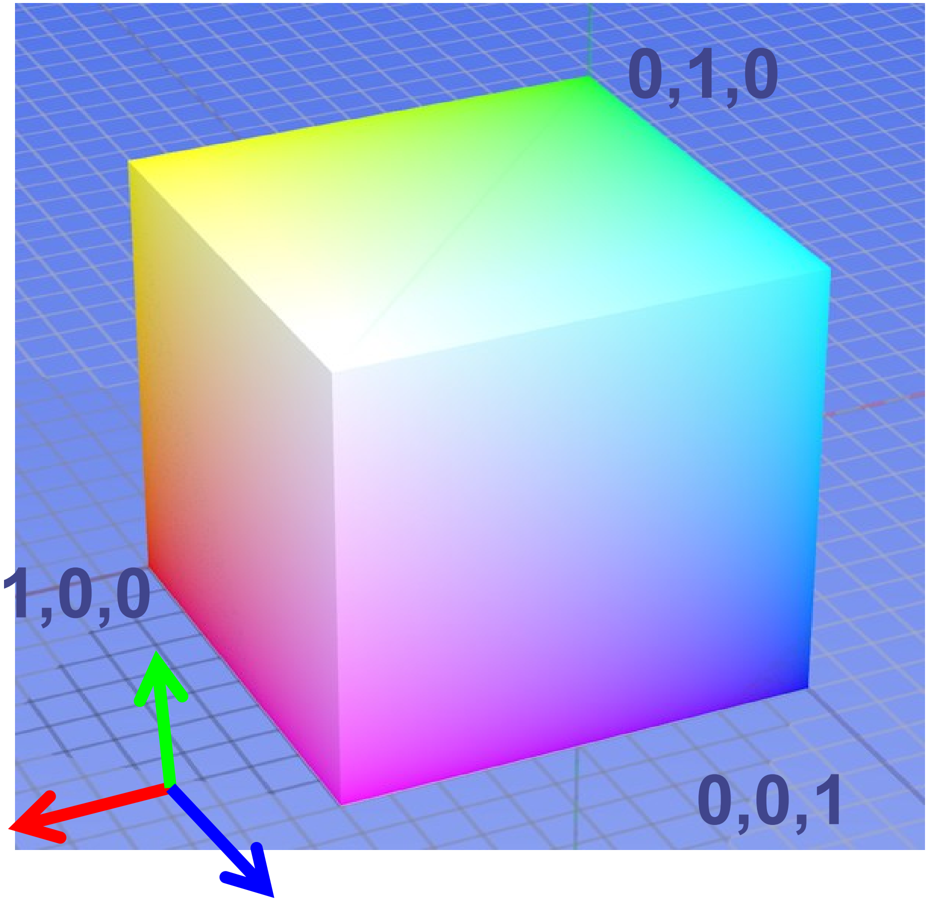
$$C + R = M + G = Y + B = White$$

- The models of color perception are firstly based on *primary colors*, **R**ed, **G**reen, **B**lue (RGB).
- Color-mixing works differently, depending whethei a camera (receiver) or a TV set (emitter) is considered: this leads to the triplet of secondary colors **C**yan, **M**agenta, **Y**ellow (CMY) which forms a basis for pigments.
- The color triangle (originated from **physiological experiments**) represents the white color in the centre, and the locus of monochromatic colors on the border.
- The **H**ue (first chromatic variable) is given by the angle between a color ***p*** and the ray connecting the corner 700 nm.
- The **S**aturation, i.e. the ratio ***wp***/***wp’,*** where ***p’*** represents the pure color perceived for that given wavelength.
- The grey scale (**I**ntensity) cannot be else than perpendicular to the triangle plane.
- This representation gave rise to a simple linear model and formulas to convert the R,G,B into the H,S,I one.



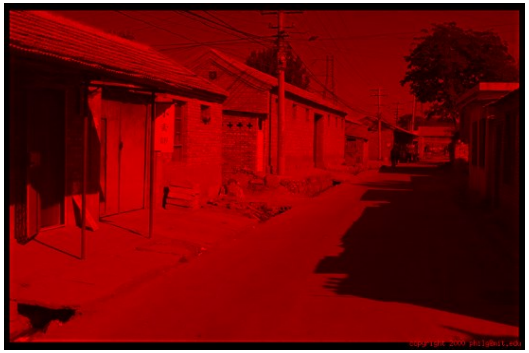
Color spaces: RGB

Default color space

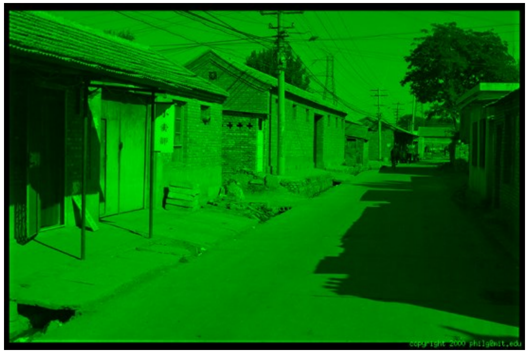


Some drawbacks:

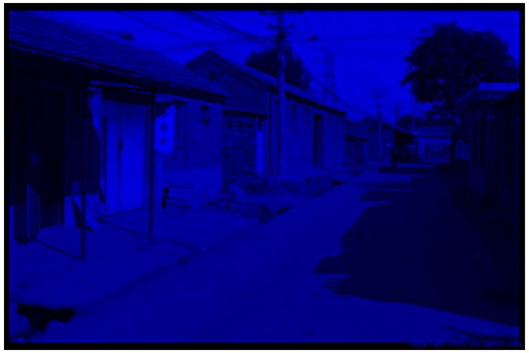
- Strongly correlated channels
- Non-perceptual



R
(G=0,B=0)



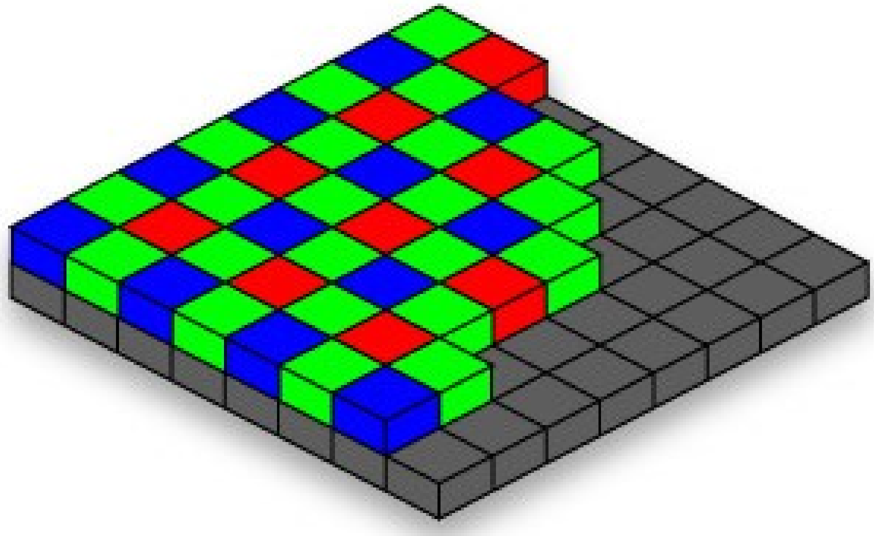
G
(R=0,B=0)

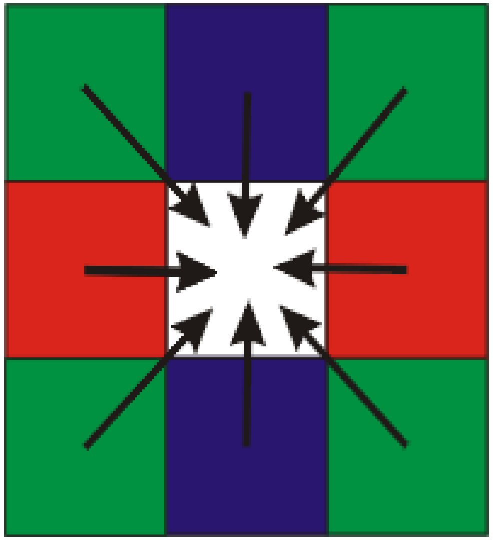


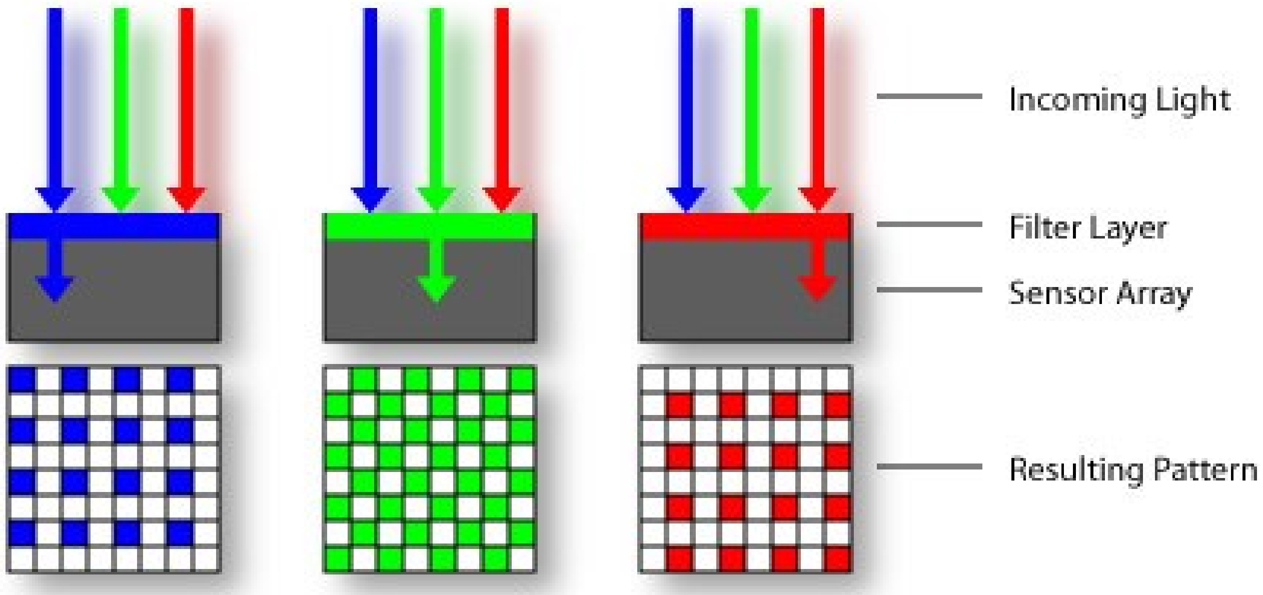
B
(R=0,G=0)

Image from: http://en.wikipedia.org/wiki/File:RGB_color_solid_cube.png

Practical Color Sensing: Bayer Grid







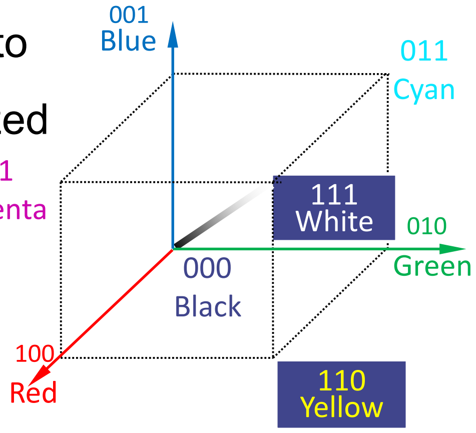
Slide by Steve Seitz

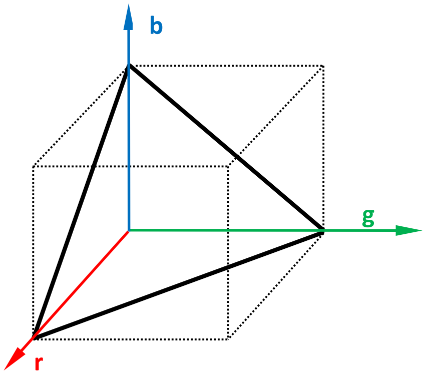
R,G,B convertion into H,S,I

- to normalize the intensity, all variables are divided by the sum R+G+B, e.g. :

  r = R/(R+G+B).

- The **I**ntensity is measured by the distance to the black. $I=\frac{R+G+B}{3}$ possibly normalized for a given Imax equal to 1.



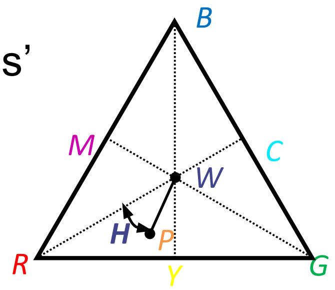


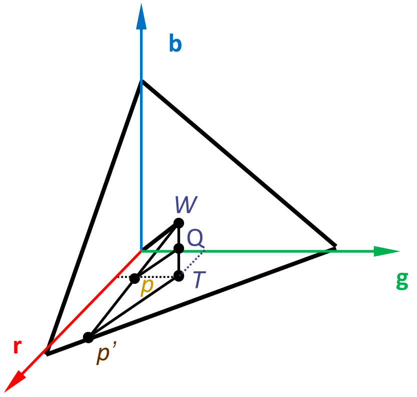
- The **H**ue, which is an angle, is given by its cosine:

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

- The **S**aturation is obtained by the Thales' theorem:

$$S=\frac{WP}{WP'}=\frac{WQ}{WT}=\frac{WT-QT}{WT}=\frac{1/3-b}{1/3}$$

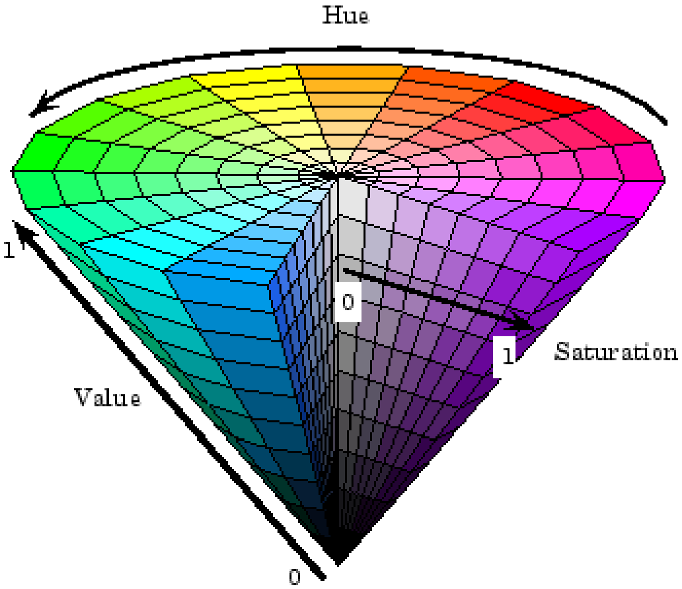


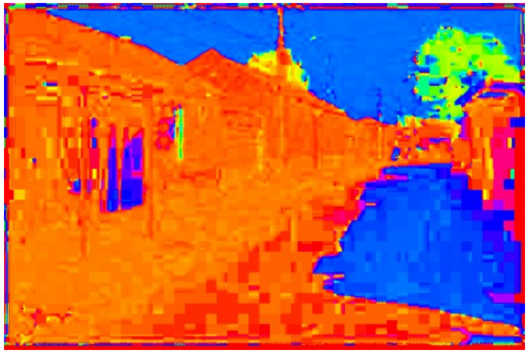


- T : projection of W in the plane (r,g), whence T = (1/3 , 1/3 , 0)
- Q : projection of P on WT, whence Q = (1/3 , 1/3 , b)

Color spaces: HSV

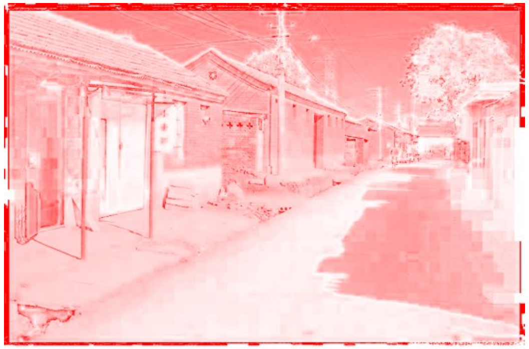
Intuitive color space





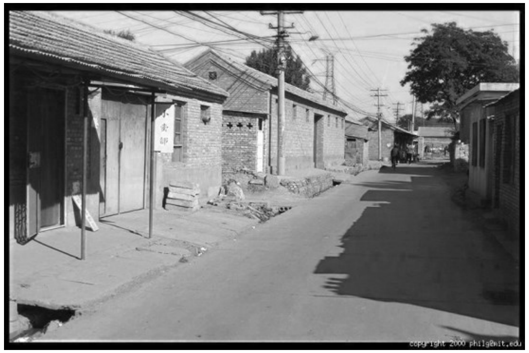
H

(S=1,V=1)



S

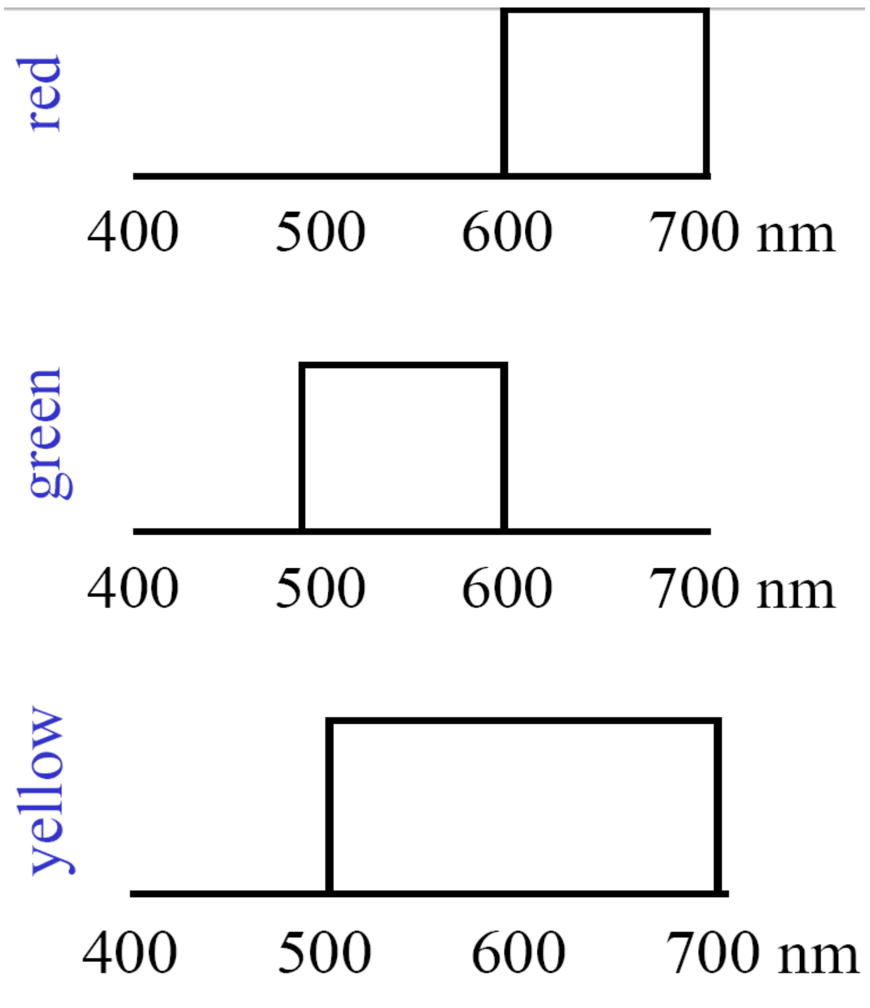
(H=1,V=1)



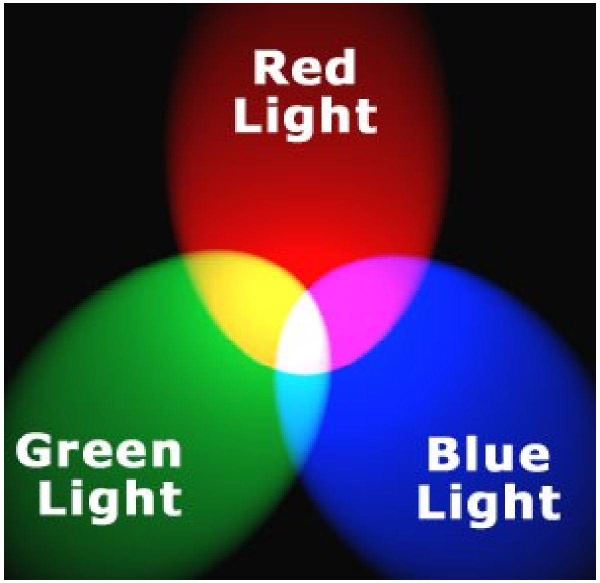
V

(H=1,S=0)

Additive color mixing



Colors combine by *adding* color spectra

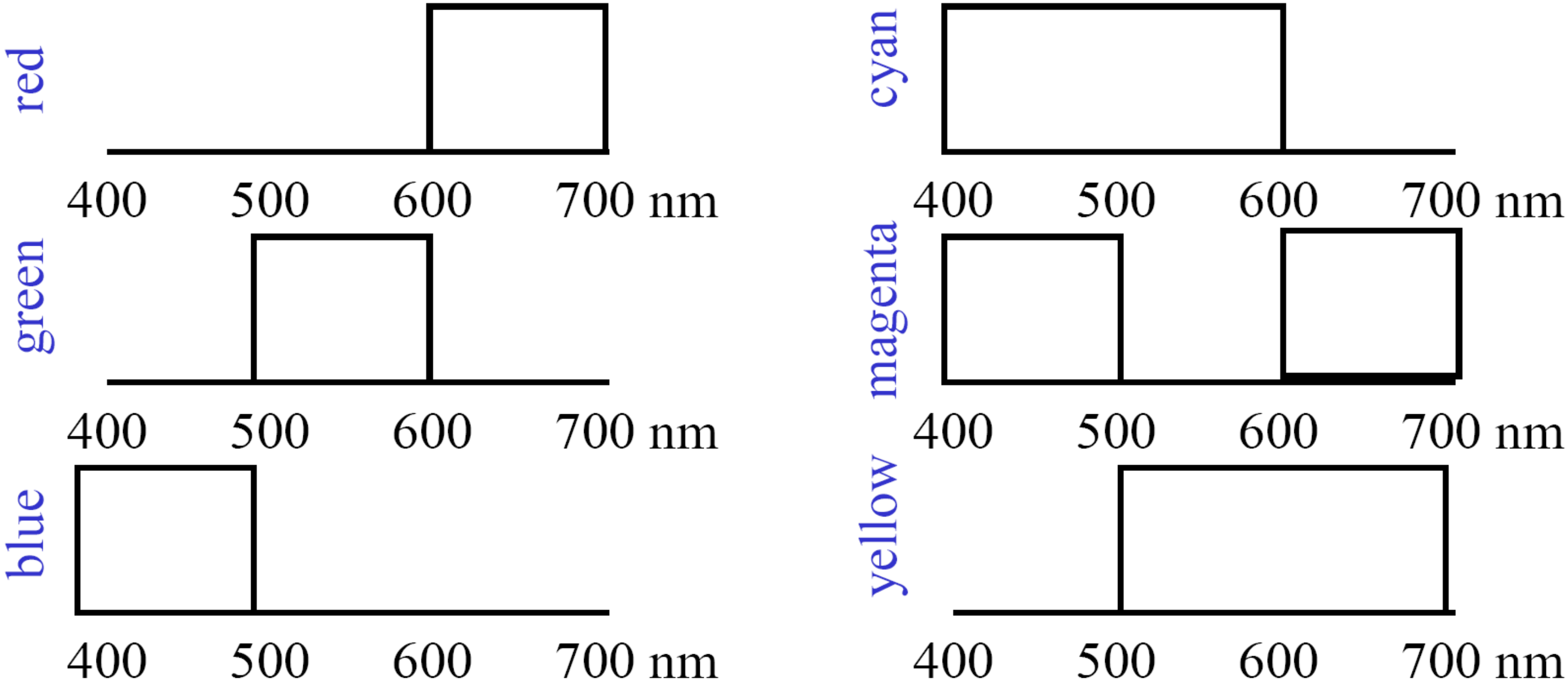


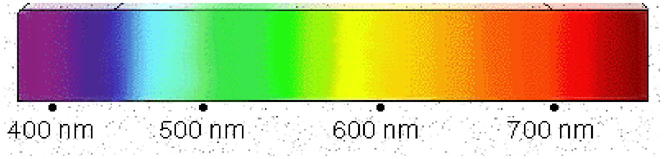
Light *adds* to black.

Source: W. Freeman

Color mixing

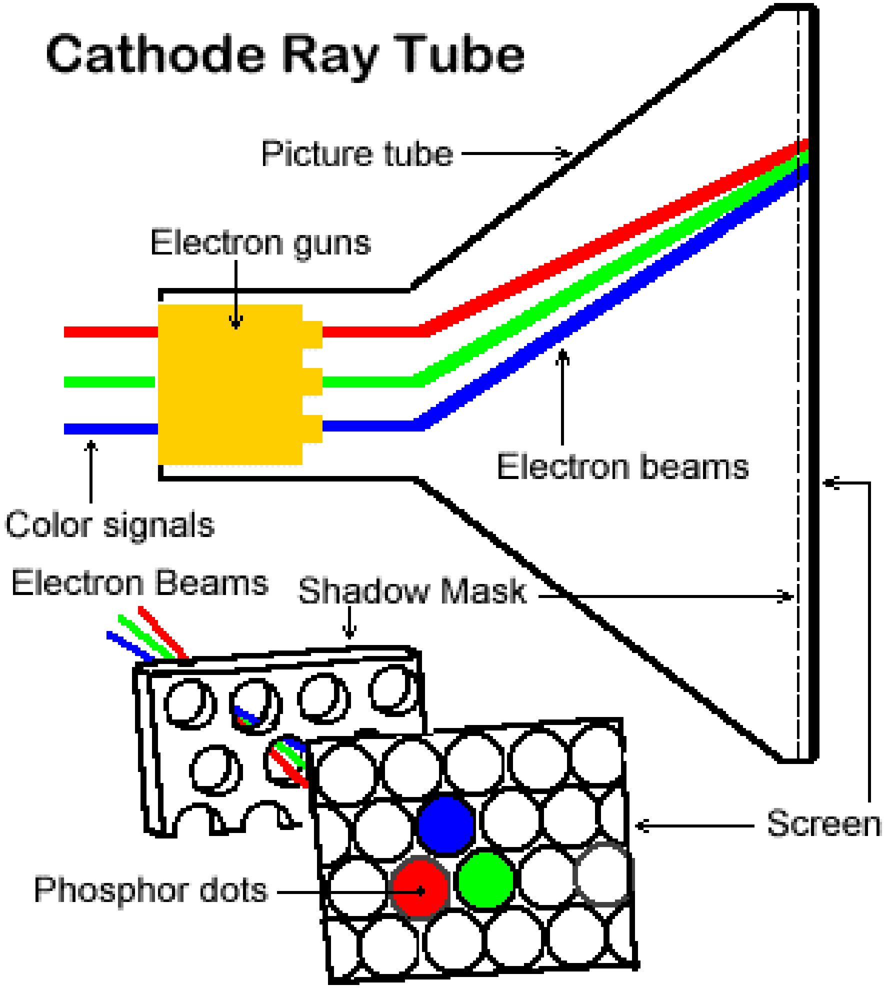
Cartoon spectra for color names:





Source: W. Freeman

Examples of additive color systems



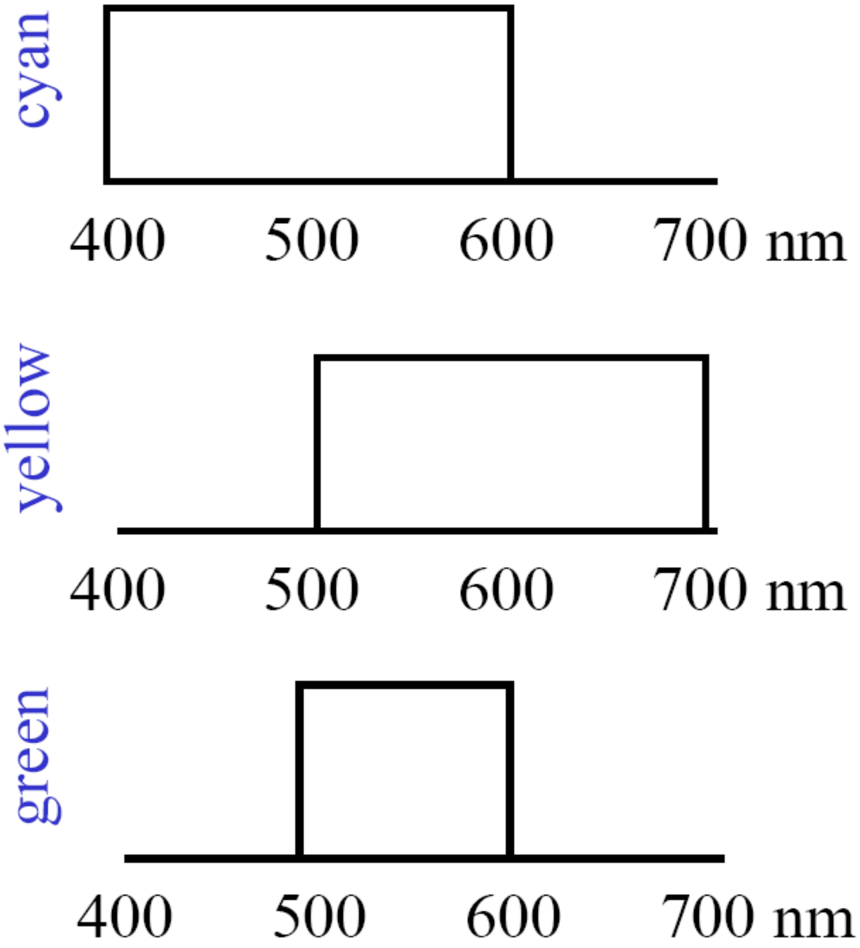
CRT phosphors

multiple projectors

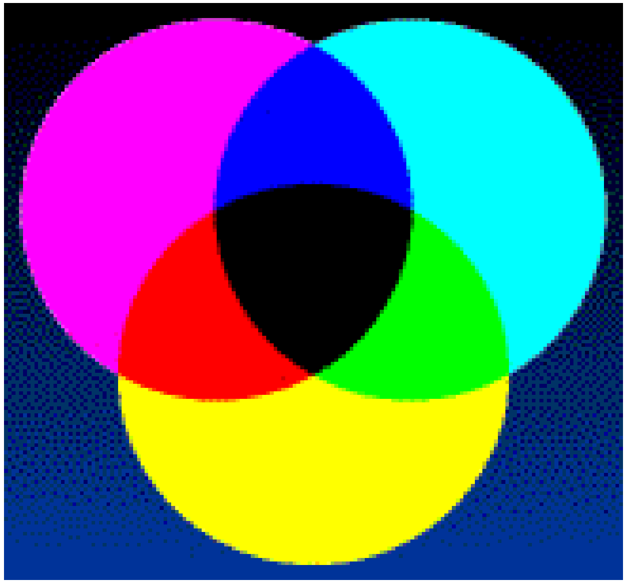
http://www.jegsworks.com

http://www.crtprojectors.co.uk/

Subtractive color mixing



Colors combine by *multiplying* color spectra.



Pigments *remove* color from incident light (white)

Source: W. Freeman

Examples of subtractive color systems

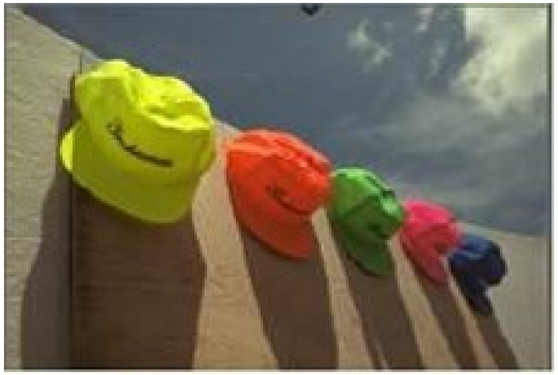
- Printing on paper
- Crayons
- Most photographic film

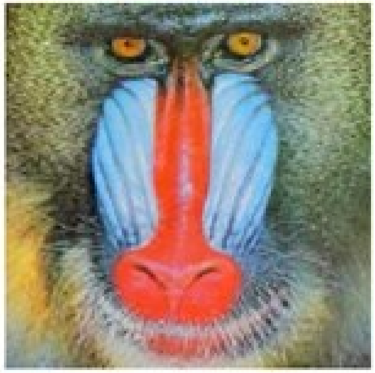
Color distribution

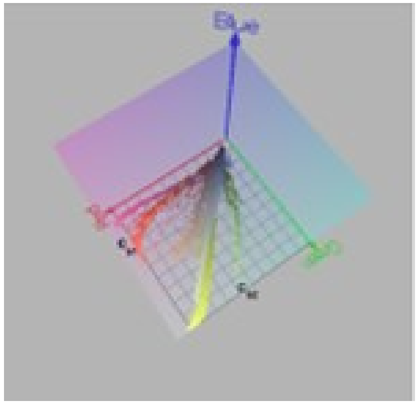
- Examples of natural color images and their color distribution in RGB space (colourSpace Software, available on:

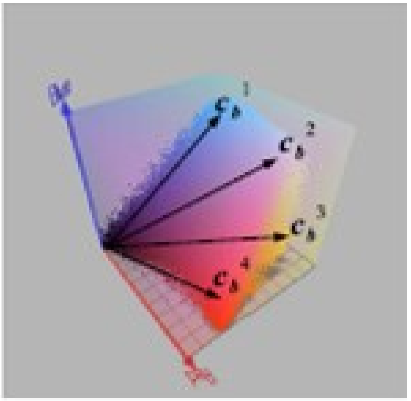
  http://www.couleur.org).

- The color components are clearly visible in the pictures containing caps, case of monochromatic items; b) less apparent in a fine grained textured image of the baboon.







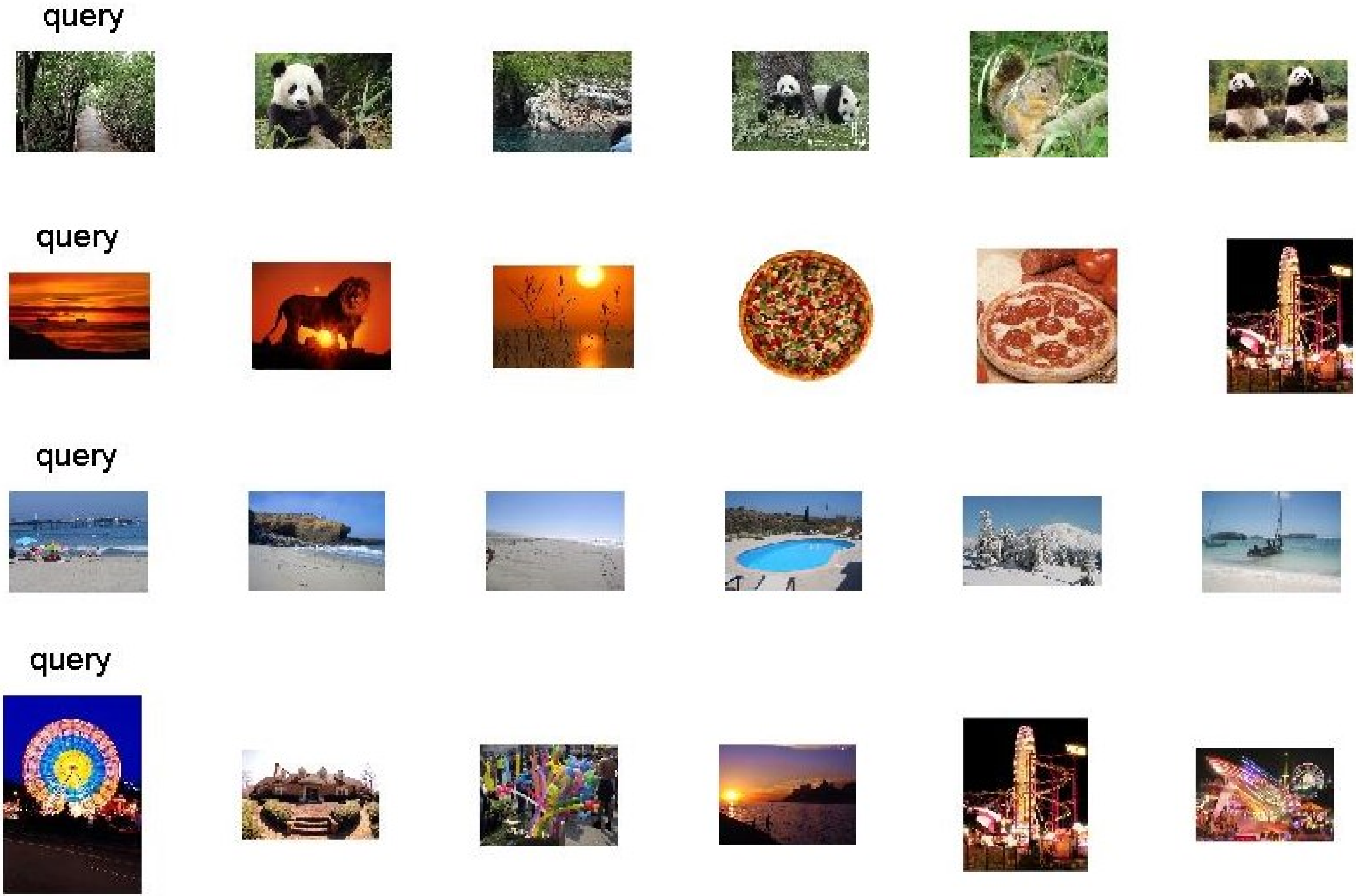


Color-based image retrieval

- Given collection (database) of images:
  - Extract and store one color histogram per image

- Given new query image:
  - Extract its color histogram
  - For each database image:
    - Compute intersection between query histogram and database histogram
  - Sort intersection values (highest score = most similar)
  - Rank database items relative to query based on this sorted order

Example retrievals: Color-based image retrieval



Example retrievals: Color-based image retrieval