ILLUMINATION GLOBALE

THEORIE ET PRATIQUE

Kadi BOUATOUCH

IRISA
RENNES

CONTENTS

LIGHT

- Light = mixture of electromagnetic waves
- Each wave : frequency, period, wavelength, energy
- Wave called *spectral component*
- Radiation : emission or transport of light energy through a medium
- Each spectral component has a color appearance
- Light described by the amount of power in each of its spectral wavelength components
- Description : spectral power distribution $\Phi(\lambda)$ (SPD)
- Sampling $\Phi(\lambda)$
- Visible spectrum : $[380nm, 780nm]$
- Number of samples : 31 samples if wavelength spacing is $10nm$
- We will see : 4 or 10 not equally spaced samples are sufficient
- SPD of the mixture of two lights = sum of the SPD's of the individual lights

- Light power Φ and its SPD $\Phi(\lambda)$ (Radiometry):

$$\Phi = \int_{380nm}^{780nm} \Phi(\lambda) d\lambda.$$

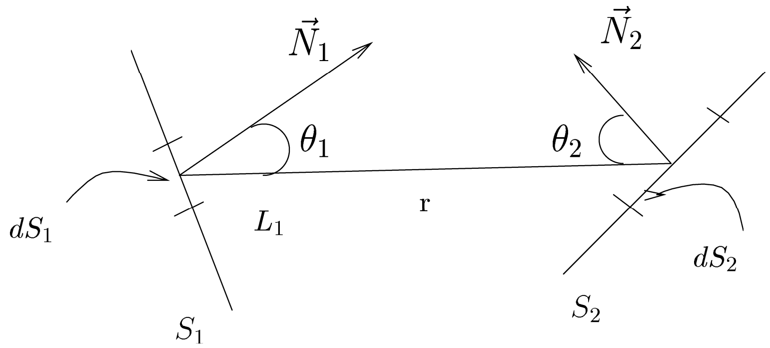
- Energetic and Luminous light powers (Phorometry):

$$\begin{aligned} \Phi_e &= \int_0^{\infty} \Phi_e(\lambda) d\lambda \\ \Phi_v(\lambda) &= 680 V(\lambda) \Phi_e(\lambda) \\ \Phi_v &= \int_0^{\infty} \Phi_v(\lambda) d\lambda, \end{aligned}$$

- $V(\lambda)$ is the sensitivity function : null out of $[380nm, 780nm]$

RADIOMETRY and PHOTOMETRY

- **Radiometry:** Measurement of quantities referring to radiation.
- **Photometry:** Measurement of quantities referring to radiation as evaluated according to a given luminous efficiency function, e.g. $V(\lambda)$.
- In the following, the expressions of all radiometric quantities are valid for light powers (Φ) as well as for each spectral component ($\Phi(\lambda)$).



- **light power or flux**: is the energy leaving a surface or impinging onto a surface per unit time.
- **radiant intensity**: is the flux leaving a surface per unit solid angle:

$$I = \frac{d\Phi}{d\Omega_1}$$

- **radiance**: is the flux leaving a surface per unit projected surface and per unit solid angle.

$$L_1 = \frac{d^2\Phi}{\cos\theta_1 dS_1 d\Omega_1}$$

  where

$$d\Omega_1 = \frac{\cos\theta_2 dS_2}{r^2}$$

- **radiant exitance** : is also called radiant emittance or radiosity. It represents the light power leaving a surface, per unit area and is given by

$$B = \frac{d^2\Phi}{dS_1} = L_1 \cos\theta_1 d\Omega_1$$

.

- **irradiance**: is the light power, per unit area, inpinging onto a surface. It is expressed as

$$A = \frac{d^2\Phi}{dS_2} = L_1 \cos\theta_2 d\Omega_2$$

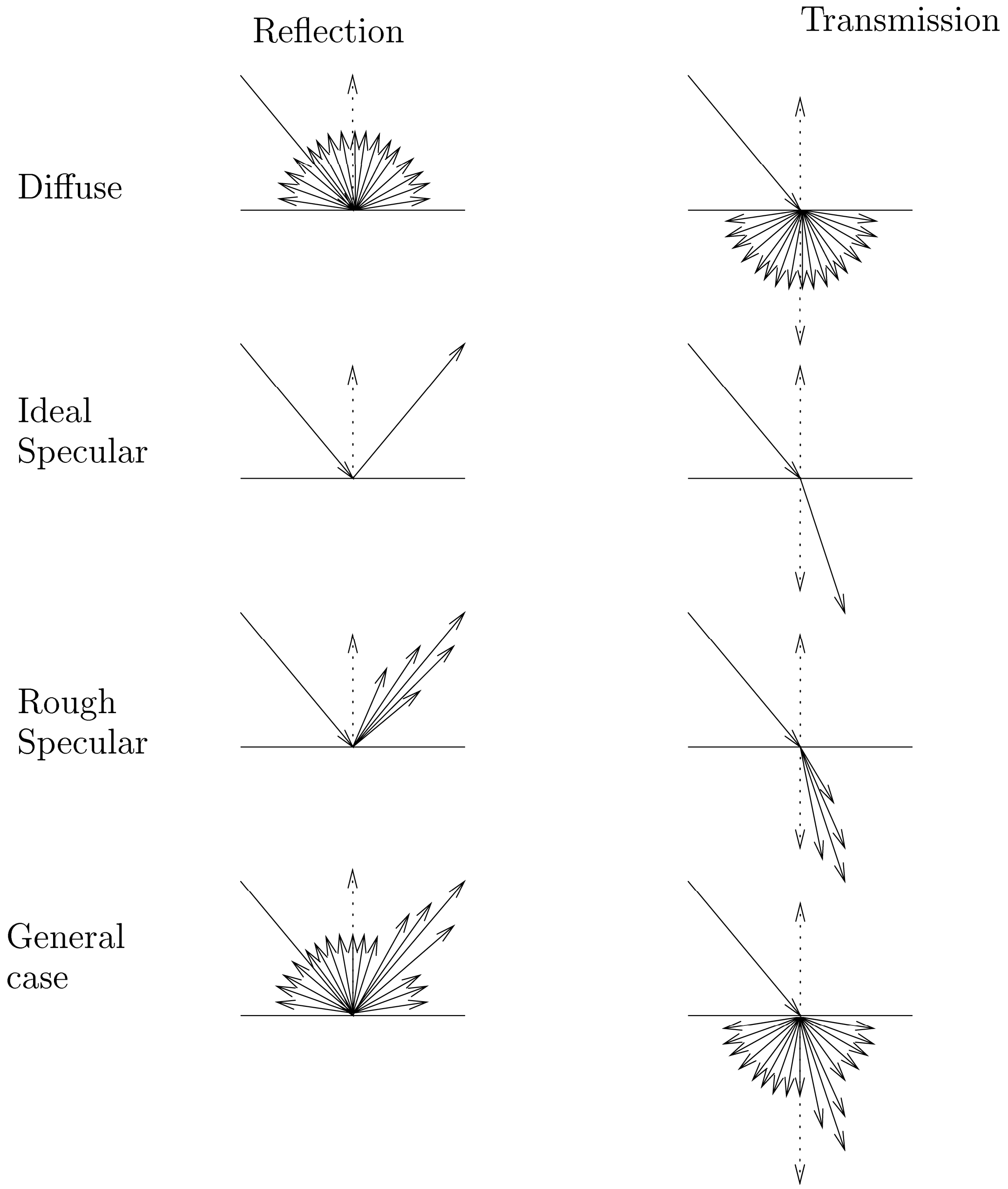
where

$$d\Omega_2 = \frac{dS_1 \cos\theta_1}{r^2}.$$

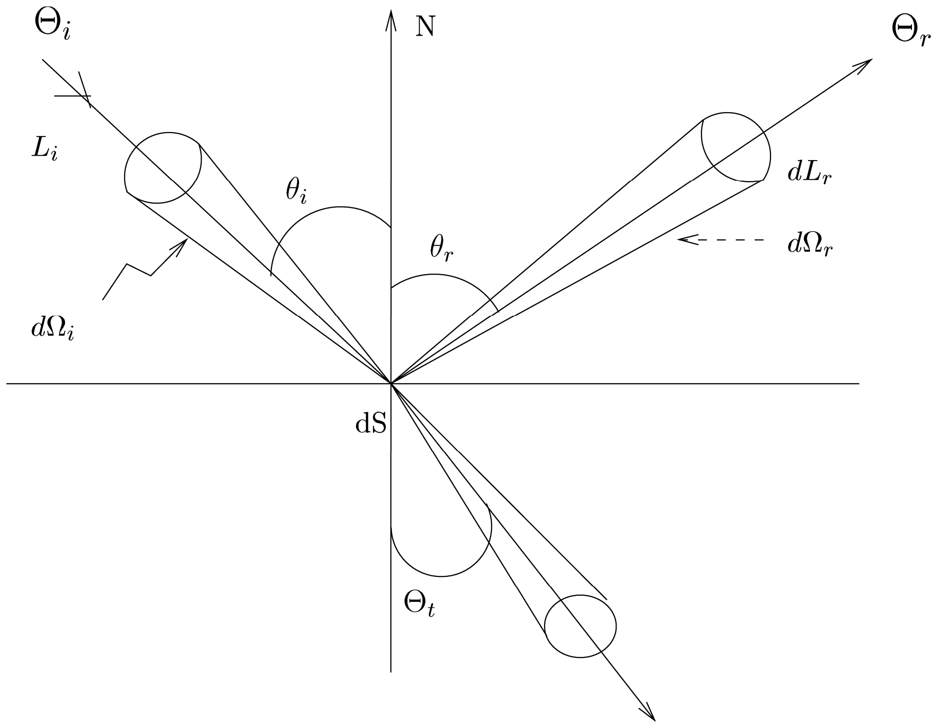
The following table recalls all these radiometric quantities.

	energetic quantity	luminous quantity
Flux	$\phi_e(\lambda)$ $(Watt)$	$\phi_v(\lambda)$ $(lumen)$
Radiant intensity	$I_e(\lambda) = \frac{d\phi_e(\lambda)}{d\Omega}$ $(Watt.strd^{-1})$	I_v $(candela)$
Irradiance or Exitance	$E_e(\lambda) = \frac{d\phi_e(\lambda)}{dA}$ $(Watt.m^{-2})$	E_v (lux)
Radiance	$L_e(\lambda) = \frac{d\phi_e(\lambda)}{d\omega dS \cos\alpha}$ $(W.strd^{-1}.m^{-2})$	L_v $(cand.m^{-2})$

REFLECTION AND TRANSMISSION



REFLECTION AND TRANSMISSION



- Reflectivity : reflected power / incident power

$$\rho = \frac{d^2\Phi_r}{d^2\Phi_i} = \frac{B_r}{A_i} = \frac{radiosity}{irradiance}$$

- Bidirectional reflectance :

$$f_r(\Theta_i, \Theta_r) = \frac{dL_r}{dA} = \frac{radiance}{irradiance}.$$

REFLECTION AND TRANSMISSION

- Or

$$f_r(\Theta_i, \Theta_r) = \frac{dL_r(\Theta_r)dS}{d^2\Phi_i(\Theta_i)}.$$

- If $d^2\Phi_i(\Theta_i)$ comes from a small light emitting surface of radianceL_i :

$$f_r(\Theta_i, \Theta_r) = \frac{dL_r}{L_i \cos\theta_i d\Omega_i}.$$

- If $d^3\Phi_r$ is the reflected power in direction Θ_r :

$$\frac{d^3\Phi_r}{d^2\Phi_i} = f_r(\Theta_i, \Theta_r)\cos\theta_r d\Omega_r.$$

- Thus :

$$\rho(\Theta_i) = \frac{d^2\Phi_r}{d^2\Phi_i} = \int_{2\pi} f_r(\Theta_i, \Theta_r)\cos\theta_r d\Omega_r.$$

- Bidirectional transmission :
  radiance dL_t in a direction of refraction Φ_t /
  irradiance for a direction of incidence Φ_i:

$$f_t(\Theta_i, \Theta_t) = \frac{dL_t}{dA} = \frac{radiance}{irradiance}.$$

REFLECTION AND TRANSMISSION

Reflection Models

- Several physics based reflection models (COOK82,HE91,WARD92)
- Focus on Cook and Torrance model
- Reflected light depends on : wavelength, incidence angle, roughness, refractive index
- Polarization of light, masking shadowing of materials
- Surface approximated by small microfacets which are assumed to be perfectly specular
- The bidirectional reflectance :

$$f_r = sR_s + dR_d \quad \text{with} \quad s + d = 1$$

- R_d and R_s are respectively the diffuse and specular components, d and s are the proportions of the incident light which give rise to the diffuse and specular components respectively.

REFLECTION AND TRANSMISSION

- $R_d = \frac{F(\lambda,0)}{\pi}$
- R_s accounts for roughness and masking/shadowing effects :

$$R_s = \frac{1}{4\pi} \frac{F(\lambda, \theta).D.G}{\cos\theta_i \cos\theta_r},$$

- Where :
  - $F(\lambda, \theta)$ is the Fresnel factor
  - θ_i is the incidence angle (direction D_i)
  - θ_r the reflection angle (direction D_r)
  - θ equals half of the angle $(D_i \hat{,} D_r)$
  - G accounts for the masking/shadowing effects between microfacets
  - D characterizes the roughness of a surface (Beckman function)

REFLECTION AND TRANSMISSION

Roughness

- D : microfacets distribution

$$D = \frac{1}{m^2 \cos^4 N \bullet H} e^{-[(\tan N \bullet H)/m]^2}$$

- G: masking and shadowing

$$G = \min\left(1, \frac{2(N \bullet H)(N \bullet V)}{(V \bullet H)}, \frac{2(N \bullet H)(N \bullet L)}{(V \bullet H)}\right)$$

REFLECTION AND TRANSMISSION

Fresnel factor calculation

- We can find in books, for several materials, Fresnel factor curves $F(\lambda, 0)$ for normal incidence, as well as the refraction index $\hat{n}$ for the wavelength $\tilde{\lambda} = 589$ (Sodium D lines) which corresponds to the center of the visible spectrum.
- Given these data, $F(\lambda, \theta)$ can be approximated, for each wavelength, by:

$$F(\lambda_i, \theta) = F(\lambda_i, 0)$$

$$+ \left(F(\lambda_i, \frac{\pi}{2}) - F(\lambda_i, 0) \right) \frac{F(\tilde{\lambda}, \theta) - F(\tilde{\lambda}, 0)}{F(\tilde{\lambda}, \frac{\pi}{2}) - F(\tilde{\lambda}, 0)},$$

where $F(\tilde{\lambda}, \theta)$ is given by the Fresnel formula for $\hat{n}$.

REFLECTION AND TRANSMISSION

- If the values of the refraction index are given for a certain number of wavelengths, then compute exactly $F(\lambda_i, \theta)$ with the help of Fresnel formula.
- Knowing the expression of $F(\lambda, \theta)$, we can precompute it for each sample wavelength and for different values of θ (20 seem enough). These values allow to create a look-up table, from which any $F(\lambda, \theta)$ can be computed by a simple linear interpolation.

REFLECTION AND TRANSMISSION

Transmission model

- So far, no physics-based transmission models have been proposed in the literature, but only an empirical one (HALL83).
- Rather than using an empirical transmission model, it is more realistic, for each material, to use transmittance values experimentally obtained with the help of a spectrophotometer.
- In case of ideal specular refraction, R_s is no more than $1 - F(\lambda, \theta)$, and $s = 1$.

Fresnel Formula

$F(\lambda, \theta_i)$: Fresnel Factor
magnitude of reflected wave (Maxwell)

$$\bar{\mathcal{R}} = \frac{1}{2}(\mathcal{R}_{\|} + \mathcal{R}_{\perp})$$

- non metallic materials

$$\mathcal{R}_{\|} = \left(\frac{n\cos\theta_i - \cos\theta_t}{n\cos\theta_i + \cos\theta_t}\right)^2$$

$$\mathcal{R}_{\perp} = \left(\frac{\cos\theta_i - n\cos\theta_t}{n\cos\theta_i + n\cos\theta_t}\right)^2$$

- metallic materials

$$\mathcal{R}_{\|} = \frac{n^2(1+k^2)\cos^2\theta_i - 2n\rho\cos\theta_i(\cos\gamma + k\sin\gamma) + \rho^2}{n^2(1+k^2)\cos^2\theta_i + 2n\rho\cos\theta_i(\cos\gamma + k\sin\gamma) + \rho^2}$$

$$\mathcal{R}_{\perp} = \frac{\cos^2\theta_i + 2n\rho\cos\theta_i(k\sin\gamma - \cos\gamma) + n^2\rho^2(1+k^2)}{\cos^2\theta_i - 2n\rho\cos\theta_i(k\sin\gamma - \cos\gamma) + n^2\rho^2(1+k^2)}$$

with

$$\rho = \sqrt{A^2 + B^2}, \quad \gamma = \frac{1}{2}\arctan\frac{B}{A},$$

$$A = 1 - \frac{(1-k^2)}{n^2(1+k^2)^2}\sin^2\theta_i, \quad B = \frac{2k}{n^2(1+k^2)^2}\sin^2\theta_i$$

COLORIMETRY

- Colorimetry : the science of measuring color based on the physical properties of light and the psychovisual properties of the human visual system.
- Maxwell tried to generate a large set of colors, by mixing three standard lights called color primaries.
- Result : most of the colors of the visible spectrum could be reproduced by combining only the three color primaries: Red, Green, Blue.
- These three color primaries must be linearly independent.
- They correspond to three different wavelengths.
- For a monitor, these three color primaries do not correspond to monochromatic lights.
- They act as a basis of a vector space called also *color space*.
- The coordinates of a color in this space are called *trichromatric components* or *tristimulus values*.

COLORIMETRY

- The trichromatic components P_i of a light of spectral distribution $E(\lambda)$ is given by :

  $$P_i = \int_{380nm}^{780nm} E(\lambda)\sigma_i(\lambda)d\lambda,$$

  where the $\sigma_i(\lambda)$'s are called *matching functions*.
- RGB (CIE)
  - The *Commission Internationale de l'Eclairage* (CIE) proposed in 1930, three color primaries: Red, Green and Blue.
  - In case of the RGB color space of a monitor, the three associated matching functions are $\bar{r}, \bar{g}, \bar{b}$. They depend on the used monitor.

COLORIMETRY

CIE XYZ color space

- The CIE has normalised a color space, in which the three color primaries X, Y, and Z are not physical colors.
- The advantage of this color space is that it is independent of the used display device.
- The particularity of this color space is that the Y component corresponds to the visual luminance of the spectrum and is obtained by taking into account the sensitivity of a reference observer ($\bar{y}(\lambda) = V(\lambda)$).

COLORIMETRY

- Trichromatic components X,Y and Z :

$$X = K \int_{380}^{780} E(\lambda)\bar{x}(\lambda)d\lambda$$
$$Y = K \int_{380}^{780} E(\lambda)\bar{y}(\lambda)d\lambda$$
$$Z = K \int_{380}^{780} E(\lambda)\bar{z}(\lambda)d\lambda$$

- For an absolute SPD, $K = 680\,lumen/watt$.
- For a relative SPD, K is selected such that bright white has a Y value of 100, then other Y values will be in the range of 0 to 100.
- Thus :

$$K = 100\,/\,\int E_w(\lambda)\bar{y}(\lambda)d\lambda$$

where $E_w(\lambda)$ is the SPD for any standard white light source (D6500).

COLORIMETRY

- The CIE standard chromaticity coordinates x, y, z are generated by projecting the tristimulus values on the $X + Y + Z = 1$ plane so that:

$$\begin{aligned} x &= X \; / \; (X + Y + Z) \\ y &= Y \; / \; (X + Y + Z) \\ z &= Z \; / \; (X + Y + Z) \\ 1 &= x + y + z \end{aligned}$$

- A common specification for color is : Y, x, y, where Y describes the luminance of the color (response to brightness) and x, y defines a point on the chromaticity diagram.
- The chromaticity diagram gives an indication of the color independent of its brightness.
- The CIE chromaticity diagram is widely used in industry for describing colors.

COLORIMETRY

Transformation from XYZ to RGB

- The transformation of a color from space RGB to space XYZ is expressed as:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

- The coefficients of the transformation matrix depend on :
  - the chromaticity coordinates $x_r, y_r, x_g, y_g, x_b, y_b$ of the phosphors of the display device,
  - the chromaticity coordinates x_w et y_w of the *white point* of the display device,
  - the luminance Y_w of this white point.

COLORIMETRY

Chromatic distance between colors

- CIELUV space

- This color space (also known as $L^*u^*v^*$), established in 1964 and adopted by the CIE in 1978.
- The three components in this space are expressed by:

$$
\begin{aligned}
L^* &= 166(Y/Y_n)^{0.5} - 16, Y/Y_n > 0.01, \\
u^* &= 13L^*(u' - u_n) \\
v^* &= 13L^*(v' - v_n)
\end{aligned}
$$

where

$$
\begin{aligned}
u' &= 4X/(X + 15Y + 3Z) \\
v' &= 9Y/(X + 15Y + 3Z) \\
u_n &= 4X_n/(X_n + 15Y_n + 3Z_n) \\
v_n &= 9Y_n/(X_n + 15Y_n + 3Z_n)
\end{aligned}
$$

- X_n, Y_n and Y_n being the trichromatic components of the reference white (ex: D6500).
- In this space the difference between two colors is expressed as:

$$\Delta E = (\Delta L^{*2} + \Delta u^{*2} + \Delta v^{*2})^{0.5}$$

- Detection of small differences

COLORIMETRY

Chromatic distance between colors

- CIELAB space

- Another system called CIELAB or L^*a^*b*, more suitable for measuring important differences between colors.
- The difference is also expressed as :

$$\Delta E = (\Delta L^{*2} + \Delta a^{*2} + \Delta b^{*2})^{0.5}$$

- Where :

$$\begin{aligned} L^* &= 166(Y/Y_n)^{0.33} - 16 \\ a^* &= 500[(X/X_n)^{0.33} - (Y/Y_n)^{0.33}] \\ b^* &= 200[(Y/Y_n)^{0.33} - (Z/Z_n)^{0.33}] \end{aligned}$$

if (X/X_n), (Y/Y_n) and (Z/Z_n) are bigger than 0.01.

TRICHROMATIC APPROACH VERSUS SPECTRAL APPROACH

- Two approaches for computing a synthetic image: trichromatic and spectral.
- Spectral : considers spectra (spectral distribution of light, spectral reflectance, transmittance and absorption, refraction index depending on wavelength...) instead of trichromatic components.
  - If $E(\lambda)$ is the incomimg light, and $f_r(\lambda)$ the bidirectional reflectance, then

$$S(\lambda) = f_r(\lambda) \times E(\lambda)$$

  - The RGB components S_R, S_G and S_B of the reflected light are obtained by:

$$\begin{aligned} S_R &= \int_{380}^{780} f_r(\lambda) \times E(\lambda)\, \bar{r}(\lambda)\, d\lambda \\ S_V &= \int_{380}^{780} f_r(\lambda) \times E(\lambda)\, \bar{v}(\lambda)\, d\lambda \\ S_B &= \int_{380}^{780} f_r(\lambda) \times E(\lambda)\, \bar{b}(\lambda)\, d\lambda \end{aligned} \qquad (1)$$

- Trichromatic : the quantities must be described by their RGB components: E_R, E_G and E_B for the incident light, and f_r^R, f_r^G and f_r^B for the reflectance function.
  - The RGB components of the reflected light are:

$$\begin{aligned} S_R &= f_r^R \times E_R \\ S_G &= f_r^G \times E_G \\ S_B &= f_r^B \times E_B \end{aligned}$$

  - These triplets are obtained by:

$$\begin{aligned} E_R &= \int_{380}^{780} E(\lambda)\, \bar{r}(\lambda)\, d\lambda \\ E_G &= \int_{380}^{780} E(\lambda)\, \bar{g}(\lambda)\, d\lambda \\ E_B &= \int_{380}^{780} E(\lambda)\, \bar{b}(\lambda)\, d\lambda \\ f_r^R &= \int_{380}^{780} f_r(\lambda)\, \bar{r}(\lambda)\, d\lambda \\ f_r^G &= \int_{380}^{780} f_r(\lambda)\, \bar{g}(\lambda)\, d\lambda \\ f_r^B &= \int_{380}^{780} f_r(\lambda)\, \bar{b}(\lambda)\, d\lambda. \end{aligned}$$

COLORIMETRY

Comparision of these two approaches

- The spectral approach leads to:

$$S_R = \int_{380}^{780} f_r(\lambda) E(\lambda)\, \bar{r}(\lambda)\, d\lambda,$$

- The trichromatic approach gives:

$$S_R = \int_{380}^{780} f_r(\lambda)\, \bar{r}(\lambda)\, d\lambda \int_{380}^{780} E(\lambda)\, \bar{r}(\lambda)\, d\lambda$$

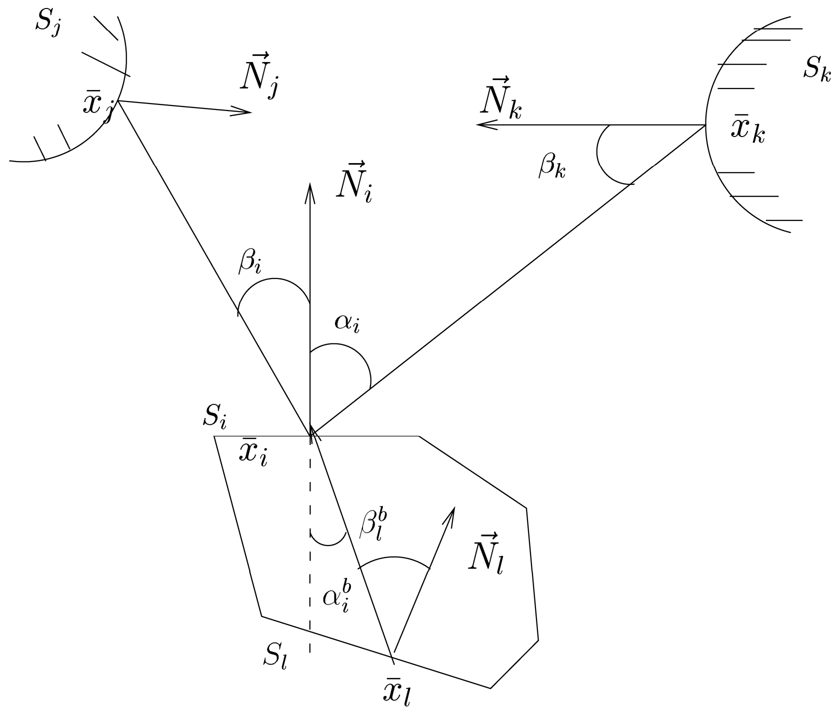
- These two last equations show that the trichromatic approach approximates an integral by the product of two integrals, which is not mathematically correct.

COLORIMETRY

Spectral sampling

- To display a light on a display device, the three trichromatic components RGB of its spectral distribution have to be calculated.
- The accuracy of this calculation strongly depends on the way the visible spectrum is sampled. It depends on both the sample values and their number.
- Meyer' s method : use The AC1C2 color space, its axes are oriented along the most dense color regions, each one having an importance that is proportional to the density of these regions.

GLOBAL ILLUMINATION MODEL



- α_i and β_i refer, respectively, to angle of incidence and angle of reflection at point $\bar{x}_i$ of a surface S_i.
- α_i^b and β_l^b refer, respectively, to angle of incidence on the back of surface S_i and angle of transmission at point $\bar{x}_l$.
- in all subscript or function argument notations, the order of the subscripts or the arguments follows the propagation of light with the source being the left-most.

GLOBAL ILLUMINATION MODEL

The global model

- $L(\bar{x_i}, \bar{x_j})$: the radiance of surface S_i at point $\bar{x_i}$ as seen from point $\bar{x_j}$ at surface S_j.
- Summing the contributions of all surfaces S_k, we have:

$$\begin{aligned} L(\bar{x_i}, \bar{x_j}) &= L^e(\bar{x_i}, \bar{x_j}) + \sum_k \int_{\Omega_{ik}} f_r(\bar{x_k}, \bar{x_i}, \bar{x_j}) L(\bar{x_k}, \bar{x_i}) \cos \alpha_i d\Omega_{ik} \\ &+ \sum_l \int_{\Omega_{il}^b} e^{-\sigma\tau} \; f_t(\bar{x_l}, \bar{x_i}, \bar{x_j}) L(\bar{x_l}, \bar{x_i}) \cos \alpha_i^b d\Omega_{il}^b \end{aligned}$$

- $f_r(\bar{x_k}, \bar{x_i}, \bar{x_j})$ and $f_t(\bar{x_l}, \bar{x_i}, \bar{x_j})$: bidirectional reflection and transmittance repectively.
- Ω_{ik} is the solid angle under which surface S_k is seen at point $\bar{x_i}$.
- Ω_{il}^b is the solid angle corresponding to the incident directions on the back of surface i, under which surface S_l is seen at point $\bar{x_i}$.
- $L^e(\bar{x_i}, \bar{x_j})$ is the radiance due to self-emittance.
- τ the traveling length of the incident ray into the transparent object, say $\bar{x_l}\bar{x_i}$.

GLOBAL ILLUMINATION MODEL

The global model

$$\begin{aligned} L(\bar{x}_i, \Theta_{out}) &= L^e(\bar{x}_i, \Theta_{out}) + \int_{2\pi} f_r(\bar{x}_i, \Theta_{in}, \Theta_{out}) L(\bar{x}_i, \Theta_{in}) \cos \alpha_i d\Omega_i \\ &+ \int_{\Omega_i^b} e^{-\sigma\tau}\ f_r(\bar{x}_i, \Theta_{in}^b, \Theta_{out}) L(\bar{x}_i, \Theta_{in}^b) \cos \alpha_i^b d\Omega_i^b \end{aligned}$$

Reflexion only :

$$\begin{aligned} L(\bar{x}_i, \Theta_{out}) &= L^e(\bar{x}_i, \Theta_{out}) \\ &+ \int_{2\pi} f_r(\bar{x}_i, \Theta_{in}, \Theta_{out}) L(\bar{x}_i, \Theta_{in}) \cos \alpha_i d\Omega_i \end{aligned}$$

GLOBAL ILLUMINATION MODEL

The global model

- As
$$d\Omega_{ik} = \frac{dS_k \cos\beta_k}{||\bar{x_k}\bar{x_i}||^2},\ d\Omega^b_{il} = \frac{dS_l \cos\beta^b_l}{||\bar{x_l}\bar{x_i}||^2}$$
- We have:
$$\begin{aligned} L(\bar{x_i},\bar{x_j}) &= L^e(\bar{x_i},\bar{x_j}) + \sum_k \int_{S_k} f_r(\bar{x_k},\bar{x_i},\bar{x_j}) L(\bar{x_k},\bar{x_i}) G(\bar{x_k},\bar{x_i}) dS_k \\ &+ \sum_l \int_{S_l} e^{-\sigma\tau}\ f_t(\bar{x_l},\bar{x_i},\bar{x_j}) L(\bar{x_l},\bar{x_i}) G'(\bar{x_l},\bar{x_i}) dS_l \end{aligned}$$
- Or:
$$\begin{aligned} L(\bar{x_i},\bar{x_j}) &= L^e(\bar{x_i},\bar{x_j}) + \sum_k \int_{\bar{x_k}\in S_k} f_r(\bar{x_k},\bar{x_i},\bar{x_j}) L(\bar{x_k},\bar{x_i}) G(\bar{x_k},\bar{x_i}) d\bar{x_k} \\ &+ \sum_l \int_{\bar{x_l}\in S_l} e^{-\sigma\tau}\ f_t(\bar{x_l},\bar{x_i},\bar{x_j}) L(\bar{x_l},\bar{x_i}) G'(\bar{x_l},\bar{x_i}) d\bar{x_l} \end{aligned}$$
- where $G(\bar{x_k},\bar{x_i}), G'(\bar{x_l},\bar{x_i})$ are purely geometric terms as
$$G(\bar{x_k},\bar{x_i}) = \frac{\cos\alpha_i \cos\beta_k}{||\bar{x_k}\bar{x_i}||^2},\ G'(\bar{x_l},\bar{x_i}) = \frac{\cos\alpha^b_i \cos\beta^b_l}{||\bar{x_l}\bar{x_i}||^2}.$$

GLOBAL ILLUMINATION MODEL

The global model

- The light occlusion effect can be accounted for by introducing a function $h(\bar{x_i}, \bar{x_j})$ taking the value 1 if point $\bar{x_i}$ is visible from point $\bar{x_j}$ and 0 otherwise.
- Thus

$$\begin{aligned} L(\bar{x_i}, \bar{x_j}) &= h(\bar{x_i}, \bar{x_j})[L^e(\bar{x_i}, \bar{x_j}) \\ &+ \sum_k \int_{\bar{x_k} \in S_k} f_r(\bar{x_k}, \bar{x_i}, \bar{x_j}) L(\bar{x_k}, \bar{x_i}) G(\bar{x_k}, \bar{x_i}) d\bar{x_k} \\ &+ \sum_l \int_{\bar{x_l} \in S_l} e^{-\sigma\tau}\ f_t(\bar{x_l}, \bar{x_i}, \bar{x_j}) L(\bar{x_l}, \bar{x_i}) G'(\bar{x_l}, \bar{x_i}) d\bar{x_l}] \end{aligned}$$

- The above system of equations completely describe the light transport mechanisms between surfaces.
- The knowledge of $L(\bar{x_i}, \bar{x_j})$ is sufficient to describe the spatial distribution of the light radiating from surface S_i.

RADIOSITY

- Surfaces perfectly diffuse.
- Thus

$$f_r(\bar{x_k}, \bar{x_i}, \bar{x_j}) = R^d(\bar{x_i}), d = 1$$
$$L(\bar{x_i}, \bar{x_j}) = L(\bar{x_i})$$

- And

$$\begin{aligned} L(\bar{x_i}) &= L^e(\bar{x_i}) \\ &+ R^d(\bar{x_i}) \sum_k \int_{\bar{x_k} \in S_k} L(\bar{x_k}, \bar{x_i}) G(\bar{x_k}, \bar{x_i}) h(\bar{x_k}, \bar{x_i}) d\bar{x_k} \end{aligned}$$

- Or:

$$\begin{aligned} L(\bar{x}) &= L^e(\bar{x}) \\ &+ R^d(\bar{x}) \int_{\bar{y} \in S} L(\bar{y}) G(\bar{y}, \bar{x}) h(\bar{y}, \bar{x}) d\bar{y} \end{aligned}$$

- If we multiply by π:

$$\begin{aligned} B(\bar{x_i}) &= E(\bar{x_i}) \\ &+ R^d(\bar{x_i}) \sum_k \int_{\bar{x_k} \in S_k} B(\bar{x_k}) G(\bar{x_k}, \bar{x_i}) h(\bar{x_k}, \bar{x_i}) d\bar{x_k} \end{aligned}$$

- Or:

$$
\begin{aligned}
B(\bar{x}) &= E(\bar{x}) \\
&+ R^d(\bar{x}) \int_{\bar{y}\in S} B(\bar{y}) G(\bar{y},\bar{x}) h(\bar{y},\bar{x}) d\bar{y}
\end{aligned}
$$

- If all the surfaces are meshed so that $B(\bar{x}_i) = constant, \forall i$:

$$
B_i = \frac{1}{S_i} \int_{\bar{x}_i \in S_i} B(\bar{x}_i) d\bar{x}_i
$$

- Then we get:

$$
\begin{aligned}
B_i &= E_i + \rho_i \sum_k F_{ik} B_k \\
A_i B_i &= A_i E_i + \rho_i \sum_k F_{ik} A_i B_k \\
A_i B_i &= A_i E_i + \rho_i \sum_k F_{ki} A_k B_k \\
\Phi_i &= \Phi_i^E + \rho_i \sum_k F_{ki} \Phi_k \\
A_i F_{ik} &= A_k F_{ki}, reciprocite
\end{aligned}
$$

- Where
  - $R^d(\bar{x}) = \rho_i/\pi$
  -
$$F_{ik} = \frac{1}{\pi S_i} \int_{\bar{x_i} \in S_i} \int_{\bar{x_k} \in S_k} G(\bar{x_k}, \bar{x_i}) h(\bar{x_k}, \bar{x_i}) d\bar{x_k} d\bar{x_i}$$
  - F_{ik} is called form factor.

RADIOSITY

The system of equations

$$B_i = E_i + \rho_i \sum_k F_{ik} B_k$$

- B_i : Exitance of patch i (Radiosity) ;
- E_i: self-emitted radiosity of patch i;
- ρ_i : reflectivity of patch i;
- F_{ik} : form-factor giving the fraction of the energy leaving patch i that arrives at patch k;
- N : number of patches.

CONSTANT RADIOSITY

The different steps

- Discretize the objects' scene into small patches.
- Calculate the form-factors, then the system matrix.
- Solve this system.
- Calculate the radiosity of each vertex of each patch by averaging the radiosities (either $B_i(\lambda)$ or B_R, B_G, B_G) of the patches sharing it. Divide them by π to convert them to radiances (if spectral approach,

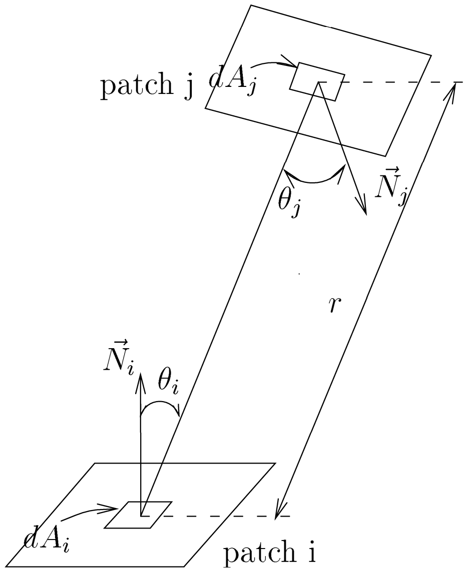
$$B_R = \int_{380}^{780} L_i(\lambda)\bar{r}(\lambda)d\lambda.$$

- Render the image by Z-buffering with Gouraud shading, or by ray tracing.

Remarks

- Solution independent of the viewpoint
- Thus, when moving the viewpoint, only the rendering step has to be run.
- Which can be handled in one second on specific graphics station. Interactivity

Form-factor calculation



- Expression :

$$F_{ij} = \frac{1}{\pi A_i} \int_{A_i} \int_{A_j} \frac{h(\bar{x_i}, \bar{x_j}) \cos \theta_i \cos \theta_j}{r^2} dA_j dA_i,$$

- The form factor between a differential element of patch i (around a point P_i) and patch j is:

$$F_{dA_iA_j} = \frac{1}{\pi} \int_{A_j} \frac{h(\bar{x_i}, \bar{x_j}) \cos \theta_i \cos \theta_j}{r^2} dA_j,$$

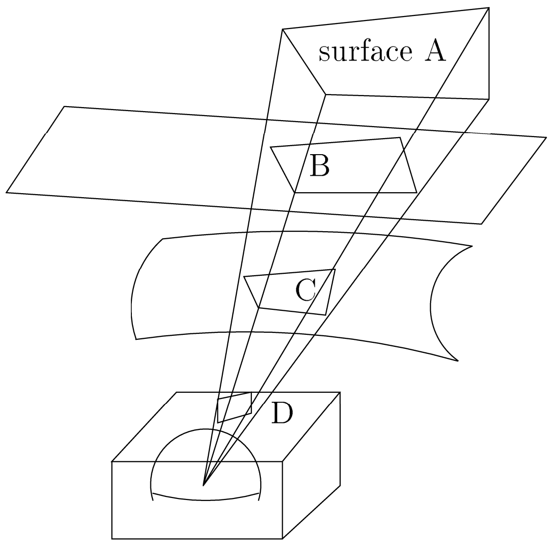
- If the two patches are far enough, this form factor is a good guess for F_{ij}.

- To compute F_{ij}, patch i is subdivided into R small elements dA_i^q and all $F_{dA_i^q A_j}$'s are evaluated.
- F_{ij} is then equal to:

$$F_{ij} = \frac{1}{A_i} \sum_{q=1}^{R} F_{dA_i^q A_j} dA_i^q$$

Projection methods

- If two patches similarly project on a given projection surface, then their form-factor (with a differential element of another patch) is thus similar.
- Find a suitable projection surface (Hemi-cube, Hemi-sphere, Plane) to simpify the form-factor calculation.

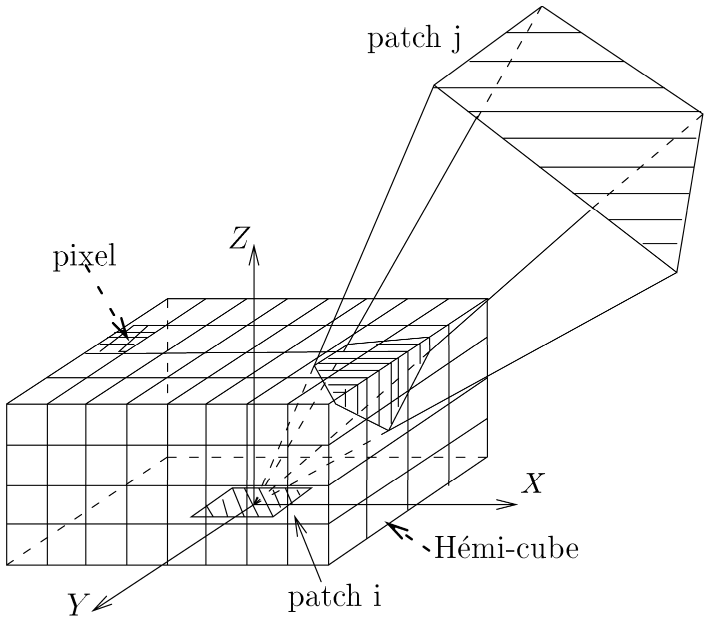


Surface Projection: HEMICUBE

Objective: compute the form factors between a patch i and all the other patches with the help of a projection surface.

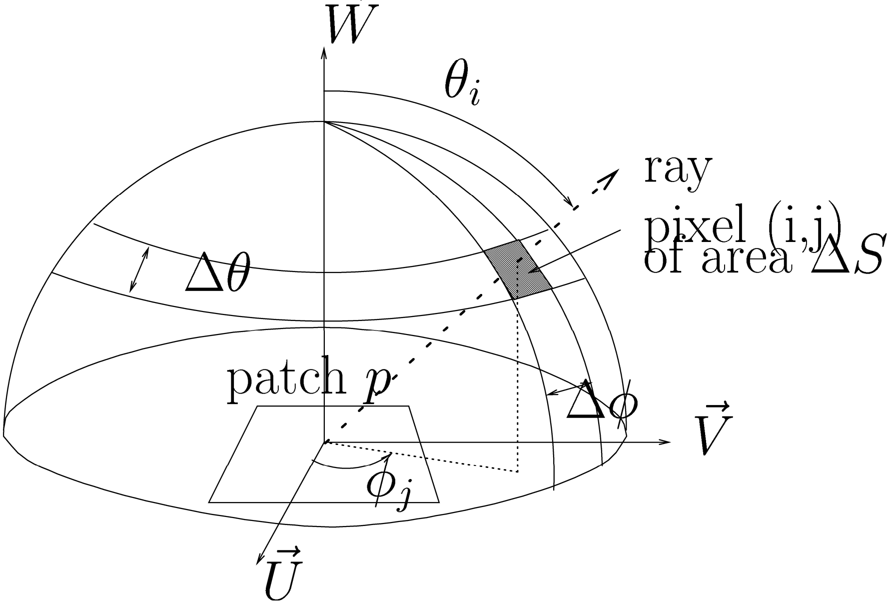
- Hemi-cube: imaginary half-cube placed at the center of the receiving patch element.
- A coordinate system is associated with this Hemi-cube, whose positive Z axis coincides with the normal to the Patch.
- Projection of the other patches onto the five faces of the Hemi-cube.
- Faces discretized into pixels

Surface Projection: HEMICUBE



- Transformation of the environment into this coordinate system.
- Clipping of the environment, Projection onto the five faces, scan-conversion.
- If two patches project on the same pixel of the hemi-cube, then Z-buffering.
- An item buffer is maintained, giving for each pixel the patch seen from the origin of the coordinate system.
- A delta form-factor is found for a diferential element dA_i to a pixel and stored in a look-up table.
- After determining which patch A_j is visible at each pixel on the hemi-cube, a summation of the delta form-factors for each pixel occupied by patch A_j (item buffer) determines the form-factor from the patch element dA_i to patch A_j.
- Then the hemi-cube is placed around another differential element of patch A_i.
- Once all the differential elements of patch A_i have been considered, the form-factors F_{ij} are evaluated, and the hemi-cube is positioned at the center of a differential element of another patch.

Surface Projection: HEMISPHERE



- Use a hemisphere as a projection surface and ray tracing.
- A hemisphere is placed at the center of a patch p and is discretized by sampling the two polar angles θ and ϕ.
- The hemisphere is then discretized into surface elements ΔS, each one corresponds to a small form-factor called delta form-factor.
- Delta form-factor associated with a pixel:

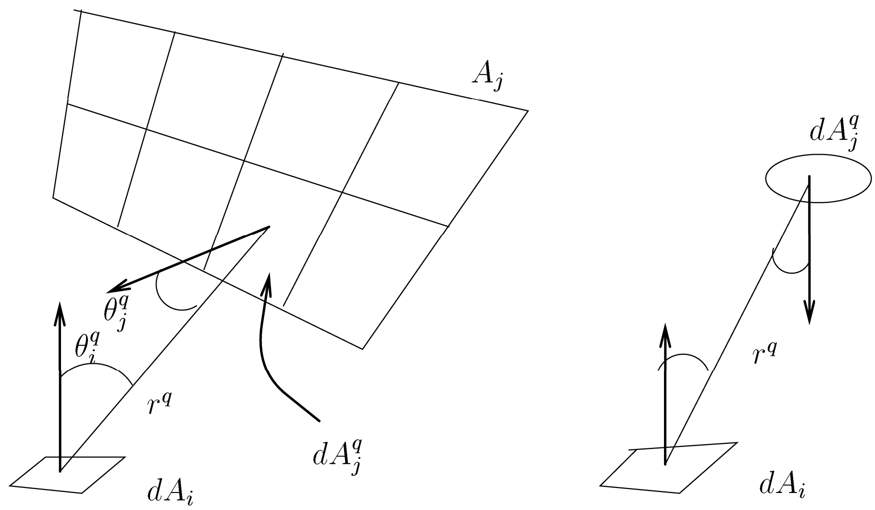
$$\Delta FF = \frac{1}{\pi} \cos\theta \sin\theta \Delta\theta \Delta\phi$$

- Visibilty : a ray is cast from the hemisphere center through each pixel $\Rightarrow$ intersection.
- The identifier of the patch containing the closest intersection point is stored in an item buffer.

- Once all the rays have been cast toward all directions (θ_i, ϕ_j), the form-factors are calculated by scanning the item buffer and summing the delta form-factors associated with the rays along which a particular patch is visible.
- This form-factor calculation technique is simpler and faster than the hemicube approach, since it avoids several processings such as polygon clipping, polygon filling and geometric transformations.

Non projection techniques

- Goal: compute the form factors for each pair (i, j) of patches.
- **Ray Tracing**



- This modified point-to-disk formula is given by:

$$F_{dA_i dA_j^q} = \frac{dA_j^q \cos\theta_i^q \cos\theta_j^q}{\pi (r^q)^2 + dA_j^q}$$

- Then

$$F_{dA_i A_j} = \sum_{q=1}^{R} \frac{dA_j^q \cos\theta_i^q \cos\theta_j^q}{\pi (r^q)^2 + dA_j^q} h(dA_i, dA_j^q),$$

- The point-to-disk formula breaks down if the distance r is small relative to the differential area.

Monte Carlo method

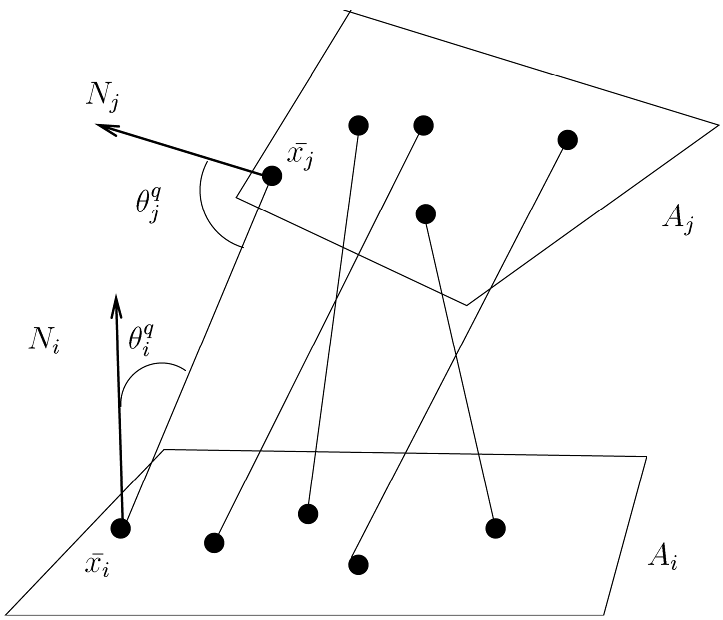


Figure 1: area-area form factor calculated by Monte Carlo

Pseudo code for Monte Carlo area-to-area form factor computation

$F_{ij} = 0$
for $k = 1$ to n do
  randomly select point $\bar{x}_i$ on the element i
  randomly select point $\bar{x}_j$ on the element j
  determine visibility between $\bar{x}_i$ and $\bar{x}_j$
  if *visible*
    compute $r^2 = (\bar{x}_i - \bar{x}_j)^2$
    compute $\cos\theta_i = \vec{r}_{ij} \bullet \vec{N}_i$
    compute $\cos\theta_j = \vec{r}_{ji} \bullet \vec{N}_j$
    compute $\Delta F = \frac{\cos\theta_i \cos\theta_j}{\pi r^2 + \frac{A_j}{n}}$
    if $(\Delta F > 0)$ $F_{ij} = F_{ij} + \Delta F$
$F_{ij} = F_{ij} * A_j$

where $\vec{r}_{ij}$ is the normalised vector from $\bar{x}_i$ to $\bar{x}_j$,
and $\vec{N}_i$ is the unit normal to element i at point $\bar{x}_i$
(and vice versa for switching i and j).

RENDERING

- The resolution of the linear system gives the radiosities $B_i(\lambda)$ or B_i^R, B_i^G, B_i^B.
- Fix the view parameters
- Trace a ray from the viewpoint toward each pixel
- P : intersection point on a patch i.
- B_P is calculated by linear interpolation.
- $$L_{pixel} = \frac{B_P}{\pi}$$

SOLVING THE SYSTEM

- Linear system :

$$KB = E, K_{ij} = -\rho_i F_{ij}, F_{ii} = 1 - \rho_i F_{ij}$$

- Solved by iterative methods:

$$B^{(k+1)} = f(B^{(k)}, B^{(k-1)}, ..., B^{(0)})$$

- $B^{(0)}$: initial guess
- Residual: $r = E - KB$
- If $r^{(k)} \approx 0$ then $B^{(k)}$ is a good estimate.

Jacobi Method

- $$\sum_j K_{ij} B_j = E_i \Rightarrow K_{ii} B_i = E_i - \sum_{j \neq i} K_{ij} B_j$$

- $\parallel r \parallel \leq \epsilon$

while not convergence **do**
  for all i **do**
    $B_i^{(k+1)} = (E_i / K_{ii}) - \Sigma_{j \neq i} K_{ij} (B_j^{(k)} / K_{ii})$

Gauss Siedel method Method

- Iteration

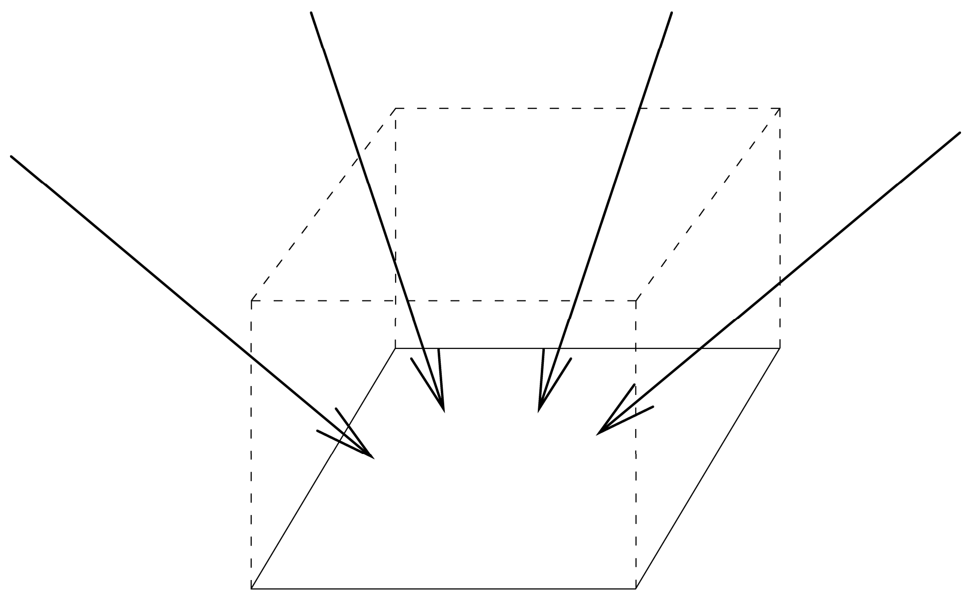
$$B_i^{(k+1)} = \frac{E_i}{K_{ii}} - \sum_{j=1}^{i-1} K_{ij} \frac{B_j^{(k+1)}}{K_{ii}} - \sum_{j=I+1}^{n} K_{ij} \frac{B_j^{(k)}}{K_{ii}}$$

- $\| B^{k+1} - B^k \|_\infty \leq \epsilon$

```
for all i do
    B_i = E_i;
while not convergence do
   for all i do
      B_i = (E_i/K_ii) + Σ_{j=1,j≠i}^n B_j K_ij/K_ii ;
```

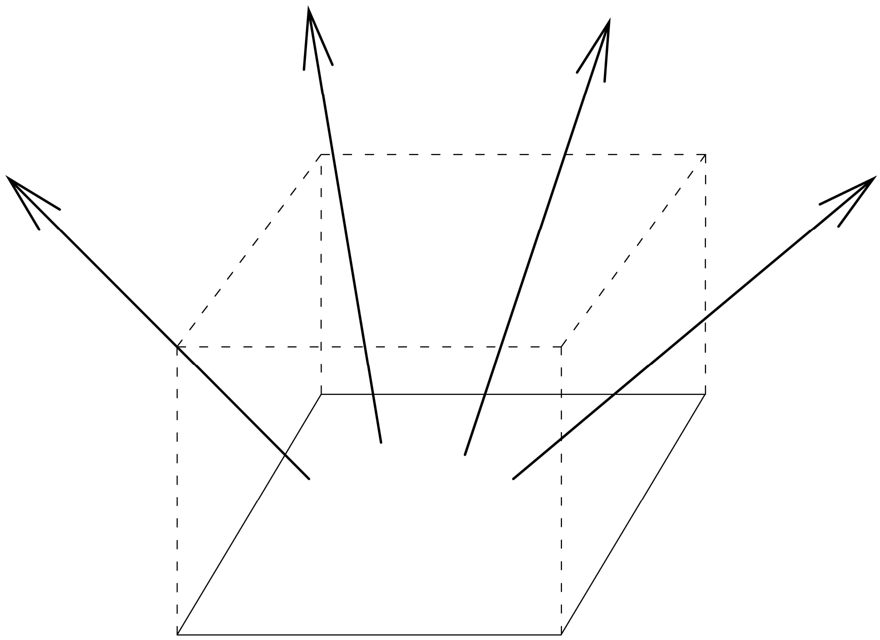
SOLVING THE SYSTEM

Complete solution



- Calculate the matrix system: memory and time complexities of $O(n^2)$.
- Resolution with Gauss-Siedel method ($K_{ij} = -\rho_i F_{ij}, K_{ii} = 1$).
- At each step, illumination from all other patches is gathered into a single receiving patch.
- The very large memory required for the storage of these form-factors limits the radiosity algorithm practically, except for hierarchical radiosity.

SOLVING THE SYSTEM:Progressive solution



- At each step, the illumination due to a single patch is distributed to all other patches within the scene. $B_{i \leftarrow j} = \rho_j B_i F_{ji} \frac{A_i}{A_j}$
- In the first steps, the light source patches are chosen to shoot their energy.
- The subsequent steps will select secondary sources, starting with those surfaces that receive the most light directly from the light sources, and so on.
- Each step increases the accuracy of the result that can be displayed.

SOLVING THE SYSTEM : Progressive solution

- The convergence criterion is met if $\| \Delta B \cdot A \|_\infty$ is below a certain threshold.
- This threshold could be a certain percentage of the sum of the total fluxes of the light sources.
- $\Delta B \cdot A$ is the vector of unshot fluxes.

```
for all i do
  for each wavelength do
    ΔB_i = E_i;
while not convergence do
  j = patch-of-max-delta-flux();
  for all i do
    for each wavelength do
      ΔRad = ρ_i ΔB_j F_ji A_j/A_i;
      ΔB_i = ΔB_i + ΔRad ;
      B_i = B_i + ΔRad;
  ΔB_j = 0 ;
```

Pseudo code for shooting

Progressive solution: Convergence and Ambient term

- After convergence, some residual fluxes remain unshot.
- Approximate them by an ambiant term.
- Ambiant term :$B_{ambiant} = R \Sigma_{j=1}^{N} \Delta B_j F_{*j}$
- Where F_{*j} represent the contribution of patch i to the others, and R characterises the mutiple interreflections.
- Expressions :

$$F_{*j} = \frac{A_j}{\Sigma_{k=1}^{N} A_k}$$

$$R = 1 + \rho_{ave} + \rho_{ave}^2 + \rho_{ave}^3 + ... = \frac{1}{1 - \rho_{ave}}$$

- ρ_{ave} is the average reflectivity of the scene objects :

$$\rho_{moyen} = \frac{\Sigma_{k=1}^{N} \rho_k A_k}{\Sigma_{k=1}^{N} A_k}$$

- Updated radiosity : $B_i = B_i + \rho_i B_{ambiant}$.

RADIOSITY: Texture Mapping

- Distribution of the reflectivities for each wavelength over a textured patch.
- Calculate the average reflectivity ρ_{ave} for each textured patch, and compute the radiosity solution.
- Obtention of radiosities B_{ave}.
- Take into account the texture values only at the rendering step.
- The radiosity for a pixel is then:

$$B_{pixel} = B_{ave} \times \frac{\rho_{pixel}}{\rho_{ave}}$$

- ρ_{pixel} is the reflectivity of the scene point seen by the observer through the pixel.

HIERARCHICAL RADIOSITY

- Method to mesh the surfaces so that the radiosity be constant over each nesh.
- Avoids unuseful finer meshing
- Reduce the number of form factor calculations
- Subdivide the surfaces adaptively according to a criterion.
- A surface : a hierarchy of surface elements.
- Leaf = element, Node = group of elements.
- Interaction between two nodes A and B of different levels.
- Interaction if A and B exchange constant energy.
- One form factor for each interaction.
- Advantage: compute form factors not for each pair of leaf nodes but for each interaction.
- Reduction of computation and memory storage.
- Link(A, B): when the two nodes interact.

HIERARCHICAL RADIOSITY: Data Structures

```
struct Quadnode {
      float B_g[ ] ; /* gathering radiosity at sample λ's */
      float B_s[ ] ; /* shooting radiosity at sample λ's*/
      float E[ ] ; /* self emittance at sample λ's*/
      float area;
      float ρ[ ] /* reflectivity at sample λ's;
      struct Quadnode** children; /* pointer to list of
                        four children*/
      struct Linknode* L; /* first gathering link of node */
}

struct Linknode {
      Quadnode* q; /* gathering node */
      Quadnode* p; /* shooting node */
      float F_qp; /* form factor from q to p */
      struct Linknode* next; /* next gathering link of node q
*/
};
```

HIERARCHICAL RADIOSITY: Hierarchy and Interactions

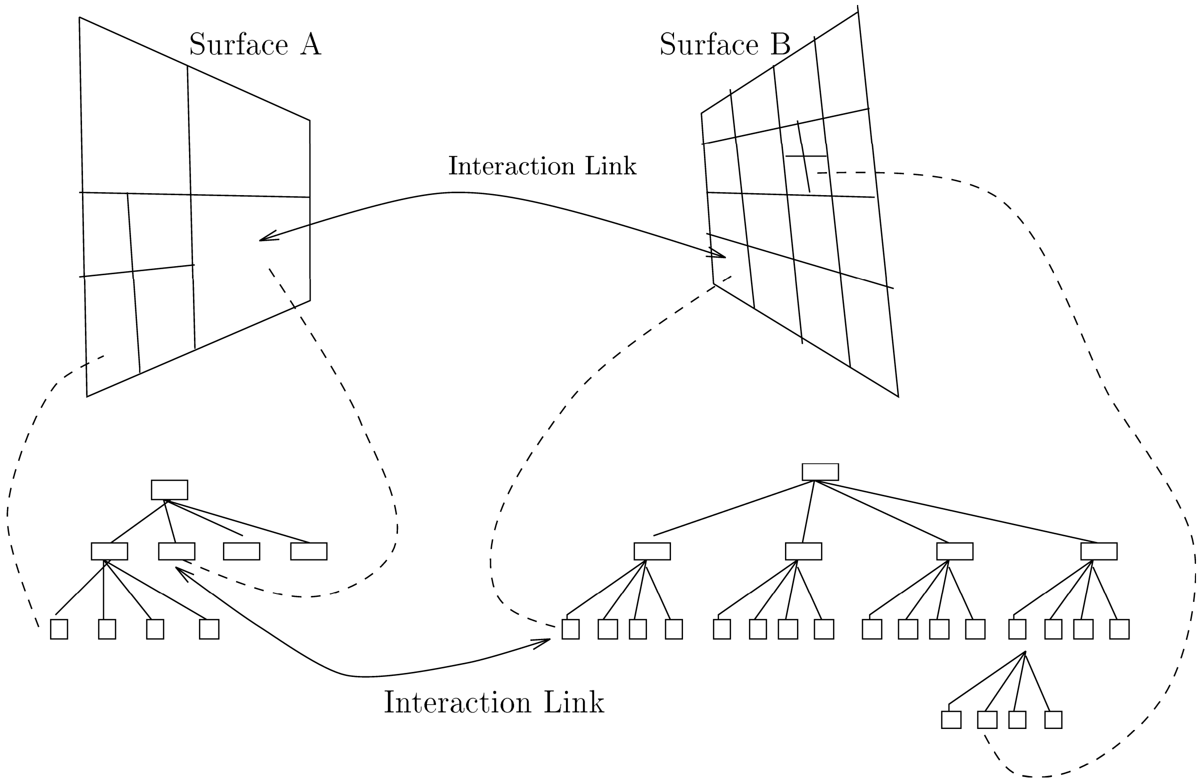


Figure 2: Hierarchy and interactions

HIERARCHICAL RADIOSITY: Refinement

- Subidivide a surface with respect to the others.
- A surface may be finely subdivided with respect to a surface A, and coarsely with respect to another surface B.

```
Refine(Quadnode *p, Quadnode *q, float F_ε)
{
    Quadnode which, r ;
    if (oracle(p, q, F_ε)) Link(p, q)
    else
        which = Subdiv(p, q);
        if (which == q )
            for (each node r of q) Refine(p, r, F_ε)
        else if (which == p )
            for (each node r of p)Refine(r, q, F_ε)
        else
            Link(p,q);
}
```

Refine pseudo code.

Solving the Hierarchical System

- **Gather**: gathers energy over each link at each receiving node. Jacobi iteration.
- **PushPull** pushes the gathered radiosity down to the children of each receiving node, and pulls the results back up the quadtrees by area averaging (decomposition and reconstruction).
- Convergence: maximum change below a threshold.

```
SolveSystem()
{
    Until Converged
        for(all surfaces p) GatherRad(p);
        for(all surfaces p) PushPullRad(p, 0.0);
}
```

Solving the Hierarchical System : Gathering

```
GatherRad(Quadnode *p)
{
    Quadnode *q;
    Link *L;

    p → B_g = 0;
    for(each gathering link L of p)
        /* gather energy across link */
        p → B_g += p → ρ * L → F_pq * L → q → B_s;
    for(each child node r of p)
        GatherRad(r);
}
```

Solving the Hierarchical System : PushPull

```
PushPullRad(Quadnode *p, float B_down)
{
    float B_up, B_tmp;

    if(p -> children == NULL)        /* p is a leaf */
        B_up = p -> E + p -> B_g + B_down;
    else
        B_up = 0;
        for (each child node r or p)
            B_tmp = PushPullRad(r, p -> B_g + B_down)
            B_up += B_tmp * (r->area / p->area);
    p -> B_s = B_up;
    return(B_up);
}
```

The Oracle

- Takes the decision to link or not two nodes.
- A link is built if F_{pq} is small enough to consider the energy contribution of p to q as small, which amounts to say that the radiosity of q due to p can be considered constant over q (**Oracle1**).
- F_{pq} is estimated as :

$$F_{pq} \approx \frac{\cos\theta}{\pi}\Omega_q,$$

where Ω_q is the solid angle whose appex is the center of p and subtented by a disk surrounding q.

```
float Oracle1(Quadnode *p, Quadnode *q, float F_ε)
{
    if (p → area < A_ε and q → area < A_ε)
        return(FALSE);
    if (EstimateFormFactor(p, q) < F_ε )
        return(FALSE);
    else
        return(TRUE);
}
```

A better Oracle

- **Oracle1** uses a geometric subdivision criterion.
- This may results in a large number of fine elements.
- More subtle to use a criterion based on the amount of energy transferred between two nodes.
- If $F_{pq} \cdot B_q \cdot A_q \leq BF_\epsilon$ then a link is established.
- Since the radiosities are not known a priori, the refinement algorithm proceeds adaptively by using another oracle **Oracle2**.

```
float Oracle2(Linknode *L, float BFε)
{
    Quadnode *p = L → p;      /* shooter */
    Quadnode *q = L → q;      /* receiver */
    if (p → area < Aε and q → area < Aε)
        return(FALSE);
    if (p → Bs == 0.0)
        return(FALSE);
    if ((p → Bs * p → Area * L → Fpq) < BFε)
        return(FALSE);
    else
        return(TRUE);
}
```

The Hierarchical Radiosity Algorithm

- **Refine** uses **Oracle1()** and establishes links at the highest levels unless the shooting surface is a light source.
- Most of these links are built even though the shooting radiosities of most the surfaces are zero.
- These links will be refined in the second pass through **RefineLink** which use **Oracle2**

The Algorithm

```
HierachicalRad(float BF_ε)
{
    Quadnode *p, *q;
    Link *L;
    int Done = FALSE;

    for (all surfaces p) p → B_s = p → E;
    for (each pair of surfaces p, q)
        Refine(p, q, BF_ε);
    while (not Done){
        Done = TRUE;
        SolveSystem();
        for (all links L)
            /* RefineLink returns FALSE if any
               subdivision occurs */
            if (RefineLink(L, BF_ε) == FALSE)
                Done = FALSE;
    }
}
```

The Hierarchical Radiosity Algorithm: RefineLink

```
int RefineLink(Linknode *L, float BF_ε)
{
    int no_subdivision = TRUE;
    Quadnode *p = L → p;      /* shooter */
    Quadnode *q = L → q;      /* receiver */

    if (Oracle2(L, BF_ε))
        no_subdivision = FALSE;
        which = Subdiv(p, q);
        DeleteLink(L);
        if(which == q)
            for (each child node r of q) Link(p,r);
        else
            for (each child node r of p) Link(r,q);
    return(no_subdivision);
}
```

GENERAL GLOBAL SOLUTION

Several approaches

- One-pass:
  - Illumination computations performed independently of the view point.
  - Fast rendering for different viewpoints.
  - Place a small relecting surface at the observer, or a non reflecting surface covering the whole virtual screen.
  - In this last case, additionnal computations are limited to the L_{ij} between the screen surface and the visible surfaces of the screen.
  - View independence.
  - Drawbacks: large memory to store data, aliasing defects due to sharp variations of specular reflections and transmissions.
  - IMMEL et al. (Siggraph, 1986), MUDUR et al. (Visualization and Computer Animation, 1990), LE SAECK et al. (Eurographics Workshop, 1990), SHAO et al. (Siggraph, 1988).

- Two-pass:
  - Global iffuse and global specular components from reflection and transmission are calculated separately.
  - Specular component is evaluated once the global one has been computed.
  - Global diffuse component: solution of a system of linear equations (similar to Radiosity)
  - Extended form-factors.
  - Global specular component: ray tracing, distributed ray tracing, Monte-Carlo.
  - Global diffuse component is view independent.
  - For different viewpoints, only the specular component has to be evaluated.
  - SILLION et al. (Siggraph, 1989, 1991), WALLACE et al. (Siggraph, 1987, 1989), HECBERT (Siggraph, 1990)

- Multi-pass:
  - Several passes.
  - Caustic effects.
  - Monte-Carlo, path tracing, ray tracing, progressive radiosity
  - SHIRLEY (Graphics Interface, 1990), CHEN et al.(Siggraph, 1991)

IMAGE DISPLAY AND VISUAL PERCEPTION

Visual perception

- The human eye converts luminance into a visual sensation, called *brightness*.
- The range of visible luminance is 10^{-6} to 5.10^4 cd/m^2.
- The visual sensation is related to the luminance, but is not linear, and depends on the ambiant level of illumination.
- Approximation: the law of sensitivity is logarithmic (Weber' law).
- Why the sensitivity varies ? Because :
  - the size of the iris varies with luminance,
  - then the sensitivity of the retina is modified.

IMAGE DISPLAY AND VISUAL PERCEPTION

Visual perception

- Take into account the sensitivity of the eye: contexts of a real scene as seen by the eye, and display device.
- Difficulty: find a function (Tumblin and Rushmeier) relating the luminance of the real world to the values to be displayed on a monitor.
- These values depend on: characteristics of the display device, and ilumination of the room containing this device.
- Overcome this difficulty: scale the image so that it fits in the color range of the monitor, and correct the non-linearities of the monitor.

IMAGE DISPLAY AND VISUAL PERCEPTION

Color clipping and scaling

- Transformation of spectral radiances into a color space (XYZ, then RGB).
- Range of RGB values very important (highlights due to specular reflections).
- Daylight is also source of important radiance variations.
- Sometimes, negative values which cannot be displayed.
- Scale and clip these components to obtain images with a maximum dynamic and a minimum loss.

IMAGE DISPLAY AND VISUAL PERCEPTION

Color scaling

- Technique used by lighting engineers.
- Consider a scene lit by artificial light sources.
- Known data: sum the luminous fluxes emitted by these sources.
- Which gives Φ_{total}.
- Assume that the total emitted flux reaches only the floor of the scene.
- Assume also that the floor is diffuse and has an average reflectivity ρ_{ave}.
- Then, the approximated average radiance of the floor is simply :

$$L_{ave} = \frac{\rho_{ave} \times \Phi_{total}}{\pi \, floor\_area}.$$

- Assume that the maximum radiance L^{max} to be displayed is fixed to twice L_{ave}, then the range for RGB components is

$$[0, 2L_R^{max}] \times [0, 2L_G^{max}] \times [0, 2L_B^{max}].$$

- Assume that the spectral distribution $L^{max}(\lambda)$ associated with L^{max} is equal to $C\Phi(\lambda)$ where C is a constant and $\Phi(\lambda) = 1 \, \forall\lambda$.
- Then :

$$L_X^{max} = K \int C \, \bar{x}(\lambda) \, d\lambda, \; L_Y^{max} = K \int C \, \bar{y}(\lambda) \, d\lambda,$$
$$L_Z^{max} = K \int C \, \bar{z}(\lambda) \, d\lambda.$$

- This yields approximatively

$$L_X^{max} \; = \; L_Y^{max} \; = \; L_Z^{max} \; = \; KC$$

.

- These XYZ components are transformed to RGB components to give $L_R^{max}, L_G^{max}, L_B^{max}$.
- Let MAXDISPLAY be equal to twice the maximum value of these three components.

- The final RGB components of the calculated radiances are computed by:

```
for each pixel
{
        if at lest one component is negative or bigger than
MAXDISPLAY
        then clip it
/* scale */
       for each component C of the pixel
         Cdisplay = (C/MAXDISPLAY)*255
}
```

IMAGE DISPLAY AND VISUAL PERCEPTION

Color clipping

- not possible to display all the calculated colors on a monitor, some of them beeing out of the gamut of the monitor, and others exceeding its range.
- Different methods of clipping (HALL89).
- Scale and clip the entire image until there are no luminances too high for display.
- Or, maintain the chromaticity and scale the luminance of the offending color.
- Or, maintain the dominant hue and luminance and desaturates the color.
- Or, clamp any color component exceeding 1 to 1.
- No method gives the best results in any case.
- Thus, set the negative value to 0, and values bigger than MAXDISPLAY to MAXDISPLAY.
- But choose an appropriate MAXDISPLAY.

IMAGE DISPLAY AND VISUAL PERCEPTION

Gamma correction

- The luminance L (Y component) produced by the phosphors of a monitor is not proportional to the input signals R, G, B.
- Non linear response: $L = kI^{\gamma}$.
- Where γ is a parameter depending on the monitor, and is about 2.3 for the three RGB channels of typical rasters.
- Assume that this law is valid for each RGB component.
- Then replace, for example, the calculated component R by $R^{1/\gamma}$.

IMAGE DISPLAY AND VISUAL PERCEPTION

Conditions for display

- Make sure that the room containing the display device is very dark , in order to avoid reflections on the screen.
- Then calibrate the monitor.
- To do this, display a totally black picture and set the brightness control so that you are just under the perception level.
- This setup must not be modified until the viewing conditions are changed.