Chapter 11

COLOR IN GRAPHICS & VISUALIZATION

Introduction

- The study of color, and the way humans perceive it, a branch of:
  - Physics
  - Physiology
  - Psychology
  - **Computer Graphics**
  - **Visualization**
- The result of graphics or visualization algorithms is a color or grayscale image to be viewed on an output device (monitor, printer)
  - Graphics programmer should be aware of the fundamental principles behind color and its digital representation

Grayscale

- **Intensity**: *achromatic* light; color characteristics removed
- Intensity can be represented by a real number between 0 (black) and 1 (white)
  - Values between these two extremes are called *grayscales*
- Assume use of d bits to represent the intensity of each pixel → $n=2^d$ different intensity values per pixel
- **Question**: which intensity values shall we represent ?
- **Answer**:
  - Linear scale of intensities between the minimum & maximum value, is not a good idea:
    - Human eye perceives *intensity ratios* rather than absolute intensity values. Light bulb example: 20-40-60W
  - Therefore, we opt for a logarithmic distribution of intensity values

Grayscale (2)

- Let Φ_0 be the minimum intensity value
  - For typical monitors: Φ_0 = (1/300) * maximum value 1 (white)
  - Such monitors have a *dynamic range* of 300:1
- Let λ be the ratio between successive intensity values
- Then we take:

$$\Phi_1 = \lambda * \Phi_0$$

$$\Phi_1 = \lambda * \Phi_1 = \lambda^2 * \Phi_0$$

$$\ldots$$

$$\Phi_{n-1} = \lambda^{n-1} * \Phi_0 = 1$$

- Given the Φ_0 of the output device, λ can be computed as:

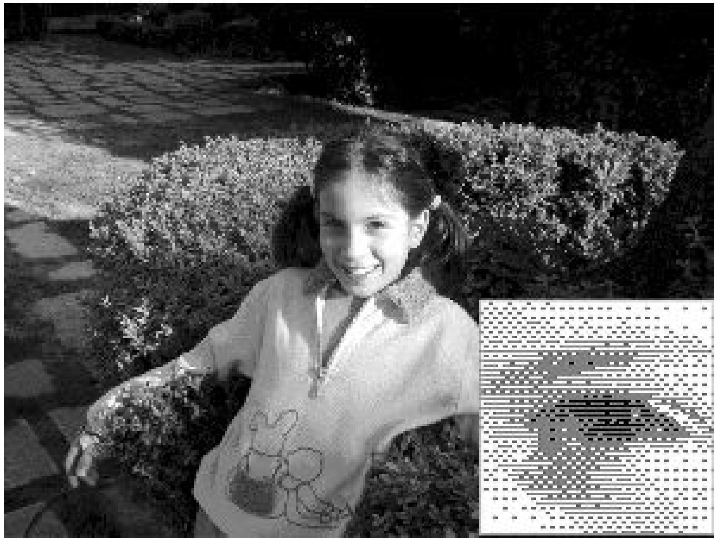
$$\lambda = (1 / \Phi_0)^{(1/n-1)} \qquad (\Lambda)$$

Grayscale (3)

- **Question**: How many intensity values do we need ?
- **Answer**:
  - if λ < 1.01 then the human eye can not distinguish between successive intensity values
  - By setting λ = 1.01 and solving (Λ) for n: $1.01^{(n-1)} * \Phi_0 = 1 \rightarrow$ $n = log_{1.01}(1/\Phi_0) + 1$
  - Since typical monitors have $\Phi_0 \sim (1/300) \rightarrow$ $n = 500$
  - On the right, we illustrate an image with n=2,4,8,16,32,64,128 and 256

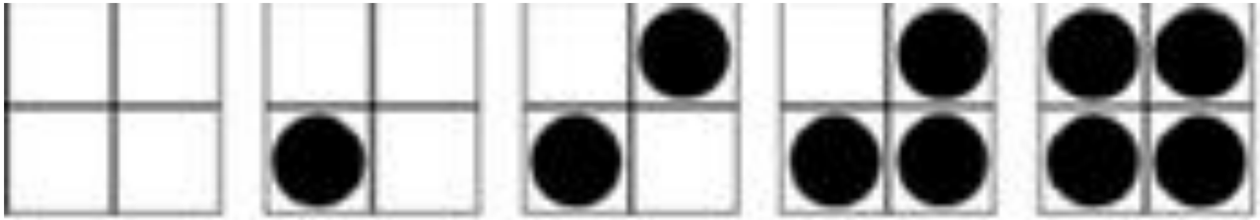
Halftoning

- *Halftoning* techniques trade (abundant) spatial resolution for grayscale (or color) resolution; opposite to antialiasing
- Halftoning techniques originate from the printing industry:
  - Black and white newspaper photographs, at a distance seem to possess a number of grayscale values, but upon closer observation one can spot the black spots of varying sizes that constitute them
  - The size of the black spots are proportional to the grayscale value that they represent



Halftoning (2)

- **Common digital approach to halftoning**: simulate the spot size by the density of "black" pixels
- Image is divided into small regions of (m x m) pixels
- Spatial resolution of regions is traded for grayscale resolution
- Spatial resolution is decreased m times in each image dimension
- Number of available grayscale values is increased by m^2
- **Example**: Consider the case of a bi-level image. Taking (2 x 2) pixel regions (m = 2) gives 5 possible final grayscale values. In general, for (m x m) regions & 2 initial grayscale values, we get m^2+1 final grayscale values



Halftoning (3)

- The above assignment of pixel patterns to grayscale values can be represented concisely by the matrix:

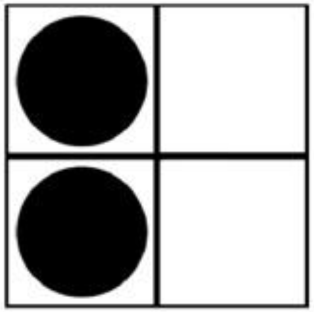
$$\begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix}$$

where a particular grayscale level k $(0 \leq k \leq 4)$ is represented by turning "on" the pixel positions of the (2 x 2) region for which the respective matrix element has a value less than k

Halftoning (4)

- Limits to the application of the halftoning technique, determined by such factors as:
  - The original spatial image resolution
  - The distance of observation

  E.g. it would make no sense to trade the full spatial resolution for a great number of grayscale levels (by making *m* equal to the image resolution)
- Sequence of patterns that define the grayscale levels must be carefully selected
- **Example:** a bad selection for grayscale level 2:



Halftoning (5)

- Useful rule: the sequence of pixel patterns that represent successive grayscale levels should be strictly incremental
  - The pixel positions selected for grayscale level i should be a subset of the positions for level j for all $j > i$
- A sequence of patterns that satisfies the quality criteria for (2x*2)* regions is:

$$\mathbf{H_2} = \begin{bmatrix} 0 & 2 \\ 3 & 1 \end{bmatrix}$$

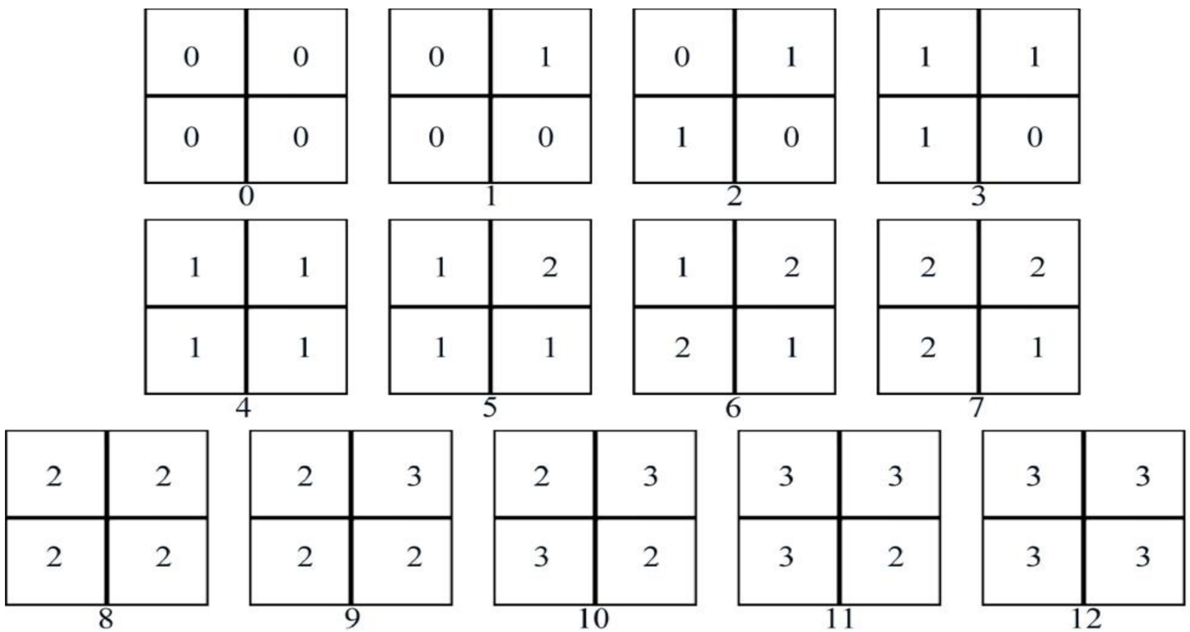
- One can recursively construct larger matrices, e.g., (4x4), (8x8) as follows:

$$\mathbf{H_m} = \begin{bmatrix} 4\cdot\mathbf{H_{m/2}} & 4\cdot\mathbf{H_{m/2}} + 2\cdot\mathbf{U_{m/2}} \\ 4\cdot\mathbf{H_{m/2}} + 3\cdot\mathbf{U_{m/2}} & 4\cdot\mathbf{H_{m/2}} + \mathbf{U_{m/2}} \end{bmatrix}, \quad m \geq 4, \quad m = 2^k$$

where $\mathbf{U_m}$ is the (m x m) matrix with all elements equal to 1

Halftoning (6)

- Halftoning can be straightforwardly extended to media which can display multiple grayscale levels per pixel
- **Example:** we can use (m x m) pixel regions to increase the number of available grayscale levels from k to $(k-1)m^2 + 1$, while reducing the available spatial resolution by m in both the x- and the y-axes:



Halftoning (7)

- Halftoning assumes that we have an abundance of spatial resolution (resolution of display medium >> resolution of image) → trade spatial for grayscale resolution
- **<u>Question:</u>** What if image and display medium have the same spatial resolutions, but the image has greater grayscale resolution than the display medium ?
- **<u>Answer 1:</u>** Simple rounding gives poor results (large amount of image information loss):

Halftoning (8)

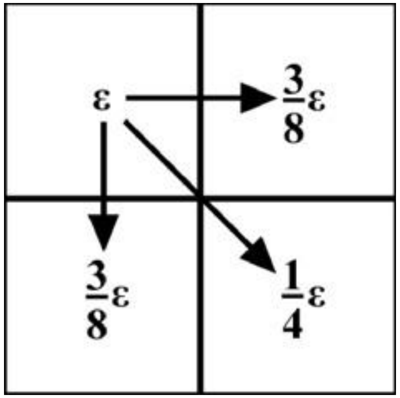
- **Answer 2:** Floyd & Steinberg proposed a method that limits information loss by propagating the rounding error from a pixel to its neighbors
- Difference ε between the image value $E_{x,y}$ and the nearest displayable value $O_{x,y}$ at pixel (x, y) is computed as:

$$\varepsilon = E_{x,y} - O_{x,y}$$

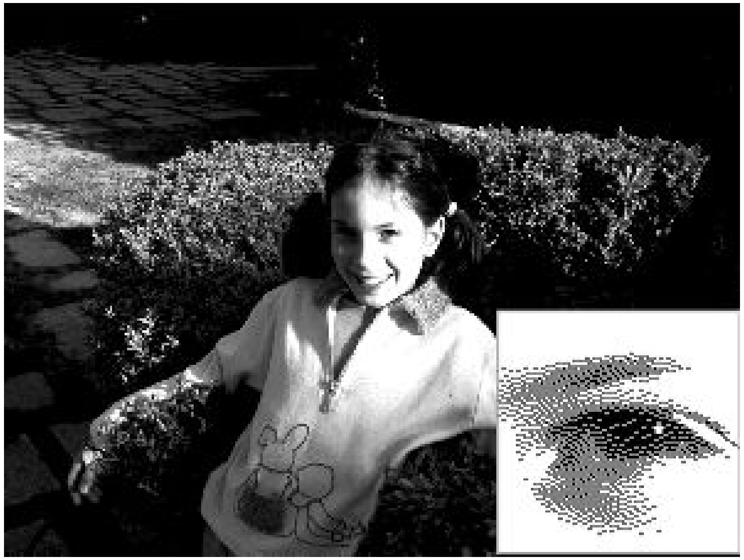
- Pixel is displayed as $O_{x,y}$ and error ε is propagated to neighboring pixels in scan-line order, as follows:

$$E_{x+1,y} = E_{x+1,y} + 3 \cdot \varepsilon / 8,$$
$$E_{x,y-1} = E_{x,y-1} + 3 \cdot \varepsilon / 8,$$
$$E_{x+1,y-1} = E_{x+1,y-1} + \varepsilon / 4$$

Halftoning (9)



- Result represents an improvement over simple rounding



Halftoning (10)

	Anti-aliasing	Halftoning	Floyd-Steinberg
Prerequisites	$I_G < D_G$	$I_S < D_S$	$I_S = D_S$ & $I_G > D_G$
Resolution gain	Spatial	Grayscale	Grayscale

- where D_G: grayscale resolution of display medium

  I_G: grayscale resolution of image

  D_S: spatial resolution of display medium

  I_S: spatial resolution of image

Gamma Correction

- Monitors have a non-linear relationship between the *input voltage* & *output* pixel intensity:

$$output = input^{\gamma}$$

where $\gamma \in [1.5 , 3.0]$ and depends on the monitor

- Input voltage values are normalized in [0, 1]
- Images, not corrected for γ, will appear too dark
- **Gamma Correction**: pre-adjust input values to ensure a linear relationship between input & output values:

$$input' = input^{1/\gamma}$$

- Giving *input'* values to the monitor, it displays the gamma-corrected image

Gamma Correction (2)

- **Left**: gamma-corrected image **Right**: non gamma-corrected image

- In practice, difficulties arise. Display systems:
  - May perform gamma-correction
  - May perform partial gamma-correction
  - May not perform gamma-correction
  - Current image formats don't store gamma-correction information → hard to deal with gamma-correction across platforms
- Gamma correction is relevant to grayscale & color images
  - For color images, it affects their intensity

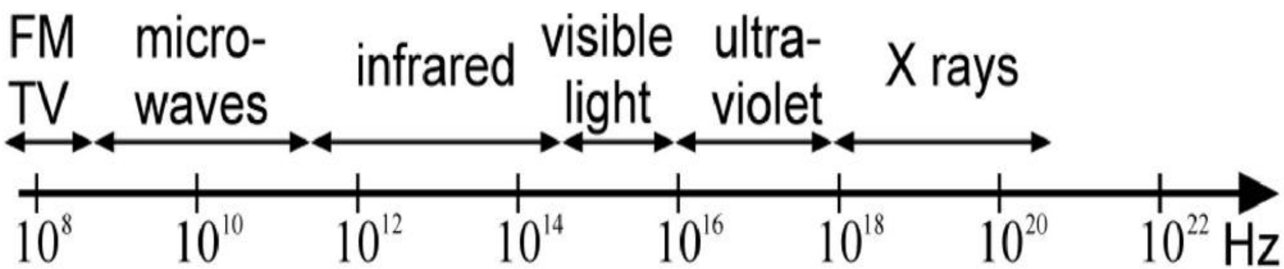
Color Models

- In a world so rich in colors, there are actually no colors!
  - Colors don’t simply exist as “deeds of light” as Goethe put it
- Colors are the product of a process that involves self-perception
- **Color Model**: a model for systematically
  - Describing
  - Comparing
  - Classifying
  - Ordering

  colors
- The simplest approach was the linear model of Aristotle:
  - Inspired by the cyclical succession of colors in the day-night continuum

Color Models (2)

- Visible colors correspond to frequencies of light:
  - They cover a small fraction of the of the electromagnetic spectrum
  - Different frequencies represent different colors
  - $4.3 \cdot 10^4$ Hz (red) to $7.5 \cdot 10^{14}$ Hz (violet)



Color models (3): Classification

A. Device – independent

- The coordinates of a color will represent a unique color
- Useful for the consistent conversion between device-dependent color models
- E.g. CIE XYZ model

B. Device – dependent

- The same color coordinates may produce a slightly different visible color value on different display devices
- E.g. RGB, CMY models
- Some models follow a device's philosophy of producing arbitrary color from the primary colors:
    i. Additive model: adds the contributions of the primaries (monitor)
    ii. Subtractive model: resembles the working of a painter / printer
        color mixing is achieved through a subtractive process

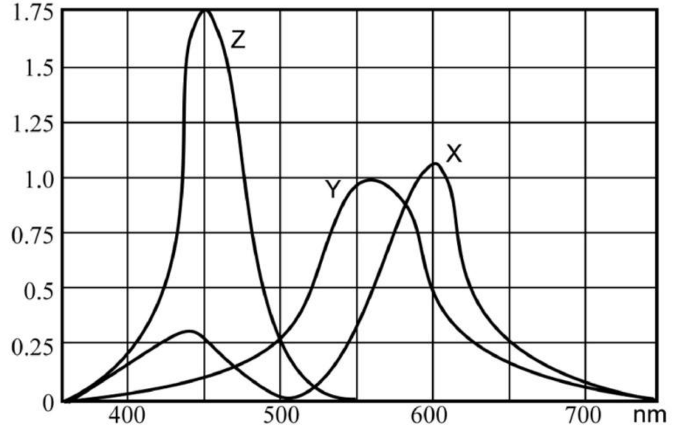
Color Models (4): other characteristics

- **Perceptual linearity:**
  - The perceived difference between 2 colors is proportional to the difference of their color values across the entire color model
- **Intuitive:** desirable
- We will examine the following color models:

1. CIE XYZ
2. CIE Yu′v′
3. CIE L*a*b*
4. RGB
5. HSV
6. CMY(K)

1. The CIE XYZ Color Model

- Grassman's 1st law:
  - Any color can be created as a linear combination of 3 basic colors
    - No combination of any subset of the basic colors can produce another
    - Analogous to the linear-independence for the basis vectors in a coordinate system
- Color representation in a 3D color space
- Color space axes are defined by the colors $\vec{\mathbf{X}}, \vec{\mathbf{Y}}, \vec{\mathbf{Z}}$
- $\vec{\mathbf{X}}, \vec{\mathbf{Y}}, \vec{\mathbf{Z}}$ are not visible colors, but computational quantities
- Mixing the basic colors in suitable proportions *X*,*Y*,*Z* produces all visible colors



1. The CIE XYZ Color Model (2)

- *X*,*Z* provide chromaticity information
- *Y* corresponds to the level of intensity (brightness)
- The basic colors form a color basis
- Other colors $\vec{\mathbf{F}}$ are expressed as linear combinations of the basis:

$$\vec{\mathbf{F}} = X \cdot \vec{\mathbf{X}} + Y \cdot \vec{\mathbf{Y}} + Z \cdot \vec{\mathbf{Z}}$$

where X,Y,Z are the color coordinates of $\vec{\mathbf{F}}$

1. The CIE XYZ Color Model (3)

- Color mixing:
  - Grassman's 2nd law
  - If $\overrightarrow{\mathbf{F_1}} = X_1 \cdot \vec{\mathbf{X}} + Y_1 \cdot \vec{\mathbf{Y}} + Z_1 \cdot \vec{\mathbf{Z}}$ and $\overrightarrow{\mathbf{F_2}} = X_2 \cdot \vec{\mathbf{X}} + Y_2 \cdot \vec{\mathbf{Y}} + Z_2 \cdot \vec{\mathbf{Z}}$ are two given colors, then their mixture is:

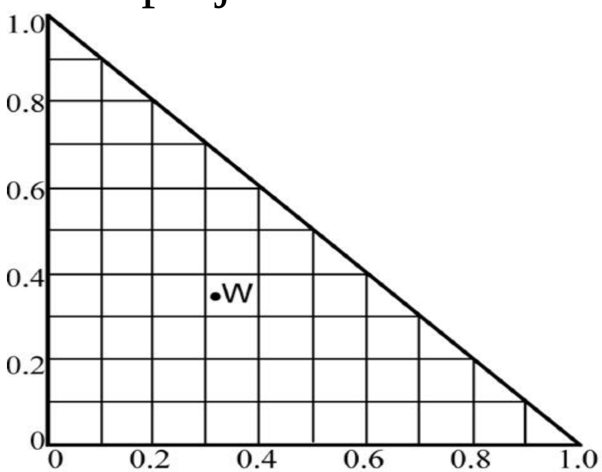
$$\overrightarrow{\mathbf{F_M}} = (X_1 + X_2) \cdot \vec{\mathbf{X}} + (Y_1 + Y_2) \cdot \vec{\mathbf{Y}} + (Z_1 + Z_2) \cdot \vec{\mathbf{Z}}$$

- Color interpolation by a factor *t* (*0*≤ *t* ≤*1*) between colors $\vec{\mathbf{F}}_1, \vec{\mathbf{F}}_2$:

$$\overrightarrow{\mathbf{F_I}} = (t \cdot X_1 + (1-t) \cdot X_2) \cdot \vec{\mathbf{X}} + (t \cdot Y_1 + (1-t) \cdot Y_2) \cdot \vec{\mathbf{Y}} + (t \cdot Z_1 + (1-t) \cdot Z_2) \cdot \vec{\mathbf{Z}}$$

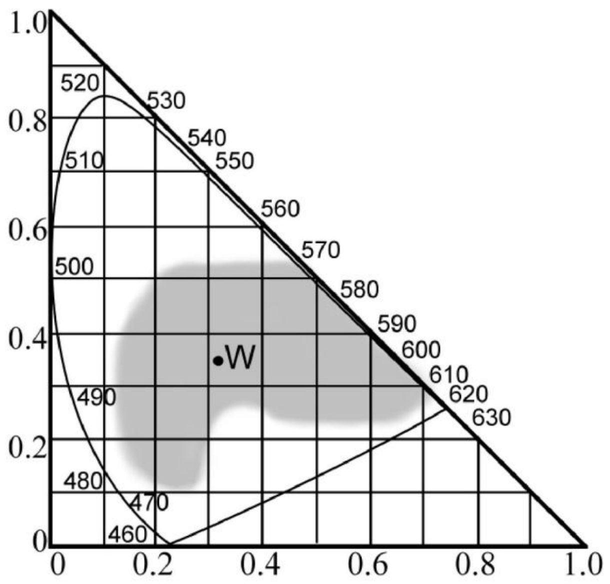
1. The CIE XYZ Color Model (4)

- The **XYZ color triangle**:
  - Created if we project the CIE XYZ model colors onto the plane $X + Y + Z = 1$
  - An arbitrary color (X,Y,Z) corresponds to the point (x, y, z) of the triangle: $x = \frac{X}{(X+Y+Z)}$, $y = \frac{Y}{(X+Y+Z)}$, $z = \frac{Z}{(X+Y+Z)}$
  - Point (x, y, z) is the intersection of vector (X,Y,Z) and the XYZ triangle
  - Since $X+Y+Z=1$, all colors of the triangle can be defined by 2 coordinates
  - The XY triangle is the projection of the XYZ triangle onto the *xy*- plane:



1. The CIE XYZ Color Model (5)

- Thus an alternative way to specify a color is **CIE Yxy**
  - Give its *x* and *y* values (or any other pair from the (*x*, *y*, *z*) triplet)
  - Give also its intensity value Y
  - Return to CIE XYZ from CIE Yxy by: $X = x \cdot \frac{Y}{y}$, $Y = Y$, $Z = (1 - x - y) \cdot \frac{Y}{y} = z \cdot \frac{Y}{y}$
- The XY triangle encompasses all visible colors
  - The shaded area represents the colors found in nature



2. The CIE Yu′v′ Color Model

- A transformation of the CIE XYZ model
- Attempts to provide perceptual linearity
- Define *u*′ and *v*′ in terms of *x* and *y* of CIE XYZ:

$$u' = \frac{4x}{-2x + 12y + 3}, \qquad v' = \frac{9y}{-2x + 12y + 3}$$

- This transformation is easily reversible
- A third component would be redundant
- A complete color specification in CIE Yu′v′ is given as a triplet (Y, u', v')
  - *Y* is the same intensity value as in CIE XYZ

3. The CIE L*a*b* Color Model

- Another transformation of CIE XYZ
- Also aims at perceptual linearity
- A device-independent color model
- Its parameters are defined relative to the *white point* of a display device
- White point:
  - The color that is displayed when all color components take their max value
  - Usually when $r = g = b = 1$
  - Is expressed in CIE XYZ as (Xn, Yn, Zn)
- CIE L*a*b* defines 3 parameters:
  - L^* for intensity (luminance)
  - a^*b^* for chromaticity

3. The CIE L*a*b* Color Model (2)

- In terms of a CIE XYZ color specification and the white point (*Xn*, *Yn*, *Zn*), the CIE L*a*b* parameters are:

$$L^* = \begin{cases} 116\sqrt[3]{Y_r} - 16, & if \quad Y_r > 0.008856, \\ 903.3 Y_r, & if \quad Y_r \le 0.008856, \end{cases}$$

$$a^* = 500(f(X_r) - f(Y_r))$$

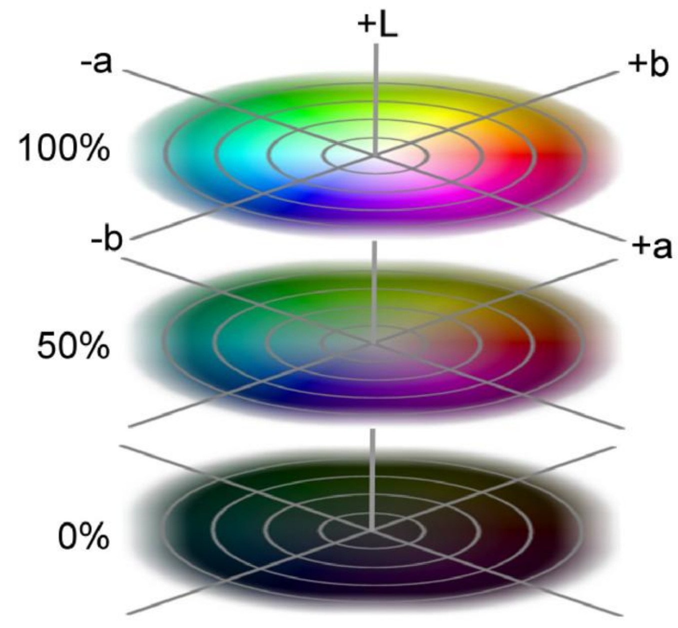
$$b^* = 200(f(Y_r) - f(Z_r))$$

where

$$X_r = \frac{X}{X_n} \quad Y_r = \frac{Y}{Y_n} \quad Z_r = \frac{Z}{Z_n},$$

$$f(t) = \begin{cases} \sqrt[3]{t}, & if \quad t > 0.008856 \\ 7.787t + 16/116, & if \quad t \le 0.008856 \end{cases}$$

- The above transformation is reversible



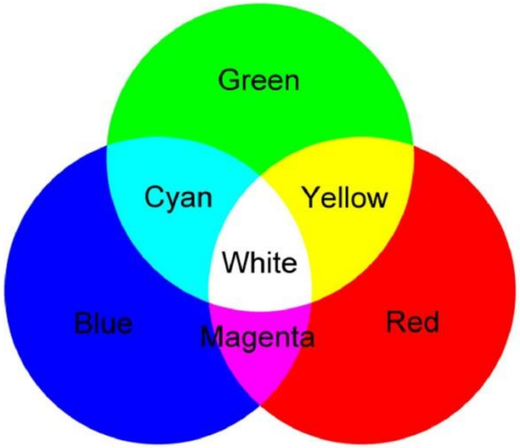
4. The RGB Color Model

- Additive color model with basic colors red, green and blue
  - Chosen because human vision is based on r, g, b color-sensitive cells
- An arbitrary color $\vec{\mathbf{F}}$ is expressed as:

$$\vec{\mathbf{F}} = r \cdot \vec{\mathbf{R}} + g \cdot \vec{\mathbf{G}} + b \cdot \vec{\mathbf{B}}$$

where $\vec{\mathbf{R}}, \vec{\mathbf{G}}, \vec{\mathbf{B}}$: the red, green, blue basis vectors

r, g, b : the color coordinates of $\vec{\mathbf{F}}$



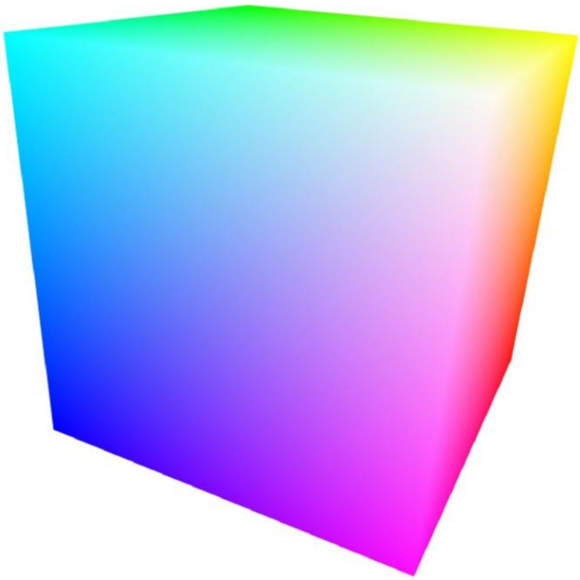
- On computer displays:
  - Colors are created using an additive method
  - Additive color mixing starts with black (no light)
  - Ends with white (the sum of all basic colors)
  - As more color is added, the result is lighter & tends to white

4. The RGB Color Model (2)

- Color scanners:
  - Work in a similar way to computer displays
  - They read the amounts of basic colors reflected from / transmitted through an object
  - Convert these readings into digital values
- The RGB model is useful for such devices due to:
  - Its additive nature
  - Its use of red, green, blue basis: visible colors, not theoretical quantities
- Color mixing and interpolation: similar to the CIE XYZ model

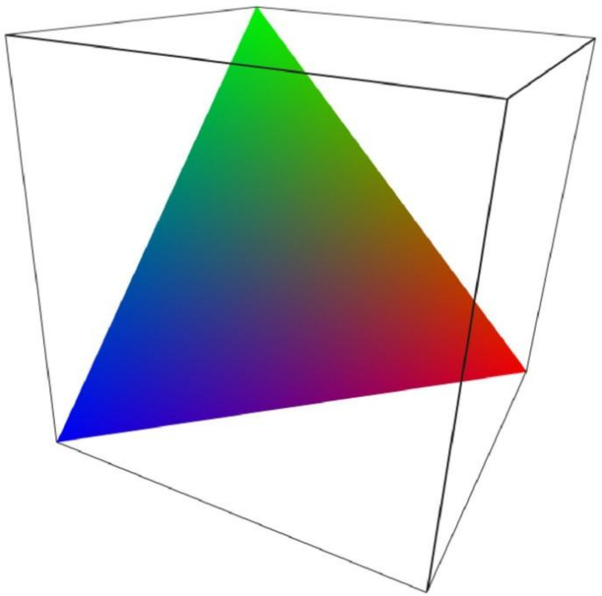
4. The RGB Color Model (3)

- **RGB cube**:
  - Is the unit cube in RGB space
  - Colors correspond to vectors from the origin (0,0,0)- the black point
  - E.g. white is (1,1,1) , green is (0,1,0)
  - The direction of a color vector defines chromaticity
  - The length of a color vector defines intensity
  - The main diagonal consists of shades of gray (from black to white)



4. The RGB Color Model (4)

- **RGB triangle**: the intersection of the RGB cube with the plane defined by the points
  - Red (1,0,0)
  - Green (0,1,0)
  - Blue (0,0,1)
- All RGB colors are mapped onto the RGB triangle
- The only information lost is intensity



4. The RGB Color Model (5)

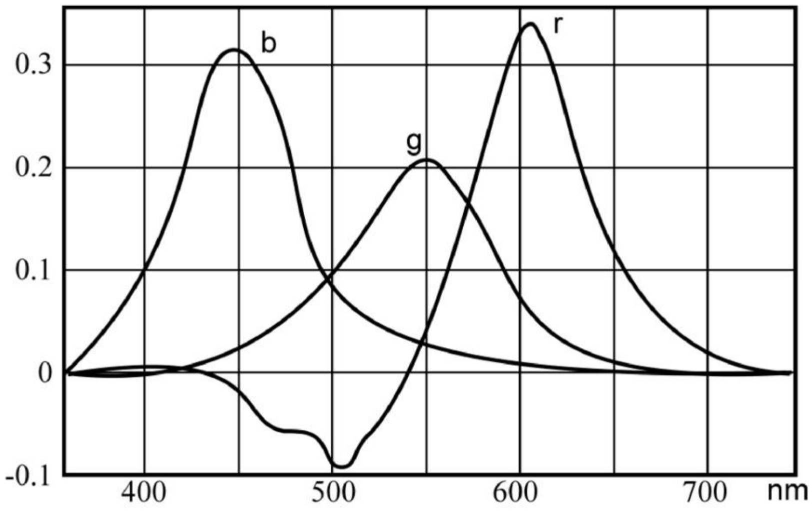
- Using the RGB triangle, we can refine the notion of chromaticity by splitting it into:

1. **Hue:**
   - Is the dominant wavelength
   - Gives a color its identity
   - All hues are found on the perimeter of the RGB triangle
2. **Saturation**:
   - Is the amount of white that is present in a color
   - Maximum at the center of the triangle
   - Minimum at its perimeter

- Colors of the same hue, but different saturation are on a line segment that connects a point on the perimeter with the triangle center
- In the RGB cube, saturation is the angle that a color vector forms with the cube diagonal

4. The RGB Color Model (6)

- Correspondence between visible colors & RGB model:
  - Portions of red, green, blue required to produce the visible colors



- RGB model is:
  - Not perceptually linear
  - Un-intuitive: it is not easy to come up with the proper RGB mix for an arbitrary color
  - Device-dependent

4. The RGB Color Model (7)

RGB is device-dependent

- The same RGB color triplet (r,g,b) will potentially produce different colors on different display devices
- Must ensure color equality when transferring color images
- Need to convert between RGB color models via an intermediate device-independent color model
- Display devices often provide a matrix **M** for the conversion:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathbf{M} \cdot \begin{bmatrix} r \\ g \\ b \end{bmatrix} \qquad \text{where} \qquad \mathbf{M} = \begin{bmatrix} X_R & X_G & X_B \\ Y_R & Y_G & Y_B \\ Z_R & Z_G & Z_B \end{bmatrix}$$

- Given the RGB to CIE XYZ conversion matrices $\mathbf{M_1}$, $\mathbf{M_2}$ of two display devices convert RGB colors between them, as follows:

$$\begin{bmatrix} r_2 \\ g_2 \\ b_2 \end{bmatrix} = \mathbf{M}_2^{-1} \cdot \mathbf{M}_1 \cdot \begin{bmatrix} r_1 \\ g_1 \\ b_1 \end{bmatrix}$$

4. The RGB Color Model (8)

Alpha color and RGB compressed modes

- The bits per pixel (bpp)
  - Is the number of bits assigned for the storage of the color of a pixel
  - Determines - the max number of simultaneous colors present in an image
    - the size of the image
  - Typically: 8 bits per color channel → 24 bpp
  - Computer words are 32 bits → the remaining 8 bits represent the **alpha value**
- Alpha color:
  - Is a quadruple $[r, g, b, a]^T$, $a \neq 0$
  - Corresponds to $[r/a, g/a, b/a]^T$
  - a represents the "area" in which the energy of the color is held
  - Can be seen as $[C, \mathrm{a}]^T$ = [energy-contribution, area-contribution], $C = r,g,b$
- The alpha representation resembles homogeneous coordinates used in projective geometry

4. The RGB Color Model (9)

Alpha color and RGB compressed modes

Example:

- Let transparent object A of alpha color $[C_A,1]^T$ be in front of transparent object B of alpha color $[C_B,1]^T$
- A is transparent so its color only contributes a fraction a_A
- We have to reduce A's area coverage
- In projective terms its contribution is $[a_A C_A, a_A]^T$
- The back object contribution is a_B of its own transparency × the portion of color energy (1- a_A) that A allows to pass through:

$$[\alpha_B(1-\alpha_A)C_B, \alpha_B(1-\alpha_A)]^T$$

- The total contribution of the 2 objects (known as *over* operator):

$$[\alpha_A C_A + \alpha_B(1-\alpha_A)C_B, \alpha_A + \alpha_B(1-\alpha_A)]^T$$

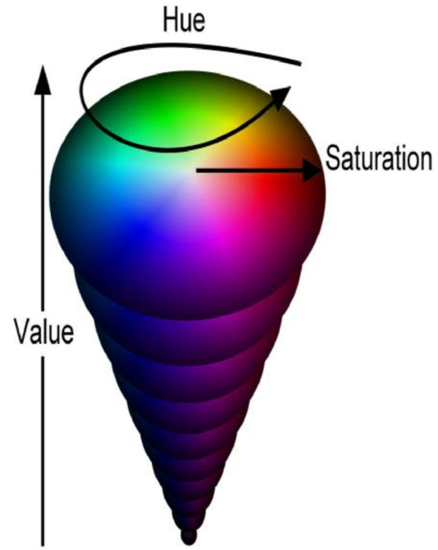
4. The RGB Color Model (10)

<u>Alpha color and RGB compressed modes</u>

- Compressed mode:
  - The size of an image is reduced by decreasing the bpp
  - Achieved by re-sampling the range of each color component
  - r:g:b:a denotes the bit allocation of the bpp into r, g, b, a
    - If 3 numbers are given → alpha is not used
    - E.g. 4:4:4:4, 5:5:5:1, 5:6:5, 3:3:2

5. The HSV Color Model

- The amounts of red, green, blue in a color indirectly control its:
  - Hue
  - Saturation
  - Intensity
- It is common to specify a color based on the above characteristics
- Artist A.H.Munsell proposed the hue-saturation-value (HSV) system
- Colors are geometrically represented on a cone



- **Hue**:
  - Arrange colors on a circle (like a color wheel) to encapsulate hue
  - Hue is the angle with respect to an initial position on the circle
  - E.g. red is at 0°, green is at 120°, blue is at 240°
  - The hue circle corresponds to a cross section of the cone

5. The HSV Color Model (2)

- **Saturation**:
  - Is max on the surface of the cone → represents pure colors with maximum "colorfulness"
  - The axis of the cone represents the min saturation (shades of gray)
- **Value**:
  - Corresponds to intensity
  - Min value (0) : absence of light (black)
  - Max value: the color has its peak intensity
  - Is represented along the axis of the cone:
    - 0 : the cone's apex
    - Max value : the center of the cone's base

6. The CMY(K) Color Model

- Subtractive color model:
  - Used during painting or printing, when colors are mixed
  - The mixing starts with white (canvas or paper)
  - As one adds color, the result gets darker & tends to black
  - E.g. if we drop cyan paint on a piece of paper, it absorbs red light

    if the paper is illuminated with white light (white = red + green + blue)

    the reflected light will be (red + green + blue) – red = cyan
- The CMY model is the complement of RGB
  - Its basic colors are cyan ($\vec{\mathbf{C}}$), magenta ($\vec{\mathbf{M}}$), yellow ($\vec{\mathbf{Y}}$)
- A color $\vec{\mathbf{F}}$ is expressed as a linear combination of the basic colors:

$$\vec{\mathbf{F}} = c \cdot \vec{\mathbf{C}} + m \cdot \vec{\mathbf{M}} + y \cdot \vec{\mathbf{Y}}$$

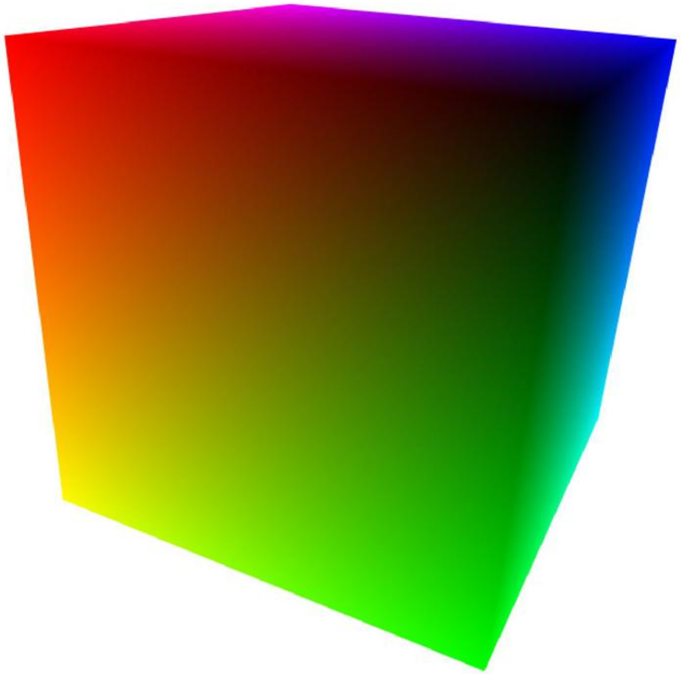
where *c*, *m*, *y*: the color coordinates of $\vec{\mathbf{F}}$

6. The CMY(K) Color Model (2)

- CMY model is perceptually nonlinear & non-intuitive (as RGB)
- Conversions between CMY and RGB:

$$\begin{bmatrix} c \\ m \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} r \\ g \\ b \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} c \\ m \\ y \end{bmatrix}$$

- The **CMY cube**:
  - Is the unit cube in CMY space
  - White appears at (0, 0, 0)
  - Black is at (1, 1, 1)
  - Other colors are in opposite vertices of those on the RGB cube



6. The CMY(K) Color Model (3)

- The CMYK color model:
  - Is a derivative of CMY that includes black
  - Black is used to offset the color composition process by the minimum components of a color $\vec{\mathbf{F}}$
- Most printers include black ink in addition to cyan, magenta, yellow
  - To avoid synthesizing black (for texts, diagrams)
  - Economize on the use of ink
  - Provide better quality of black

6. The CMY(K) Color Model (4)

- Conversion from CMY to CMYK:

$$b = \min(c, m, y)$$

$$c' = \frac{c - b}{1 - b}$$

$$m' = \frac{m - b}{1 - b}$$

$$y' = \frac{y - b}{1 - b}$$

where c', m', y', b: the components of CMYK

6. The CMY(K) Color Model (5)

- Conversion from RGB (display) to CMY (printer):
  - Both models are device-dependent
  - Should convert from RGB to a device-independent system (e.g. CIE XYZ)
  - Then convert to CMY, using the transformation matrices of the devices:

$$\begin{bmatrix} c \\ m \\ y \end{bmatrix} = \begin{bmatrix} XYZ \rightarrow \\ CMY \\ ofprinter \end{bmatrix} \cdot \begin{bmatrix} RGB \rightarrow \\ XYZ \\ ofdisplay \end{bmatrix} \cdot \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

- Summary of color models:

	Device-independent?	Perceptually linear?	intuitive?
CIE XYZ	Y	N	N
CIE Yu′v′	Y	Y	∼N
CIE L*a*b*	Y	Y	∼N
RGB	N	N	∼N
HSV	N	N	Y
CMY	N	N	∼N

Web Issues

- When making images for the web:
  - They will be viewed by a large audience, with various display systems
  - The same digital image can appear different on different display systems

1. Difference in gamma correction:
   - An image stored with different gamma correction than that of the actual display system will appear too bright or too black
   - Use an “average” gamma correction, e.g. 2.2
2. Difference in the color model:
   - Common to store images in the device-dependent RGB model
   - For the transfer of images consider one of the CIE models
   - But this has drawbacks:
     i. has an extra step of calibration
     ii. requires an expensive conversion if an semi-intuitive model is used
     iii. RGB models are widely accepted for display devices

Web Issues (2)

sRGB (standard RGB)

- Easier to handle for device manufacturers because it provides:
  - colorimetric definition of the red, green, and blue basic colors in terms of the device-independent standard CIE XYZ
  - a gamma of 2.2
  - precisely defined viewing conditions

- Device – independent
- Useful in consumer electronics (e.g. digital cameras)

High Dynamic Range (HDR) Images

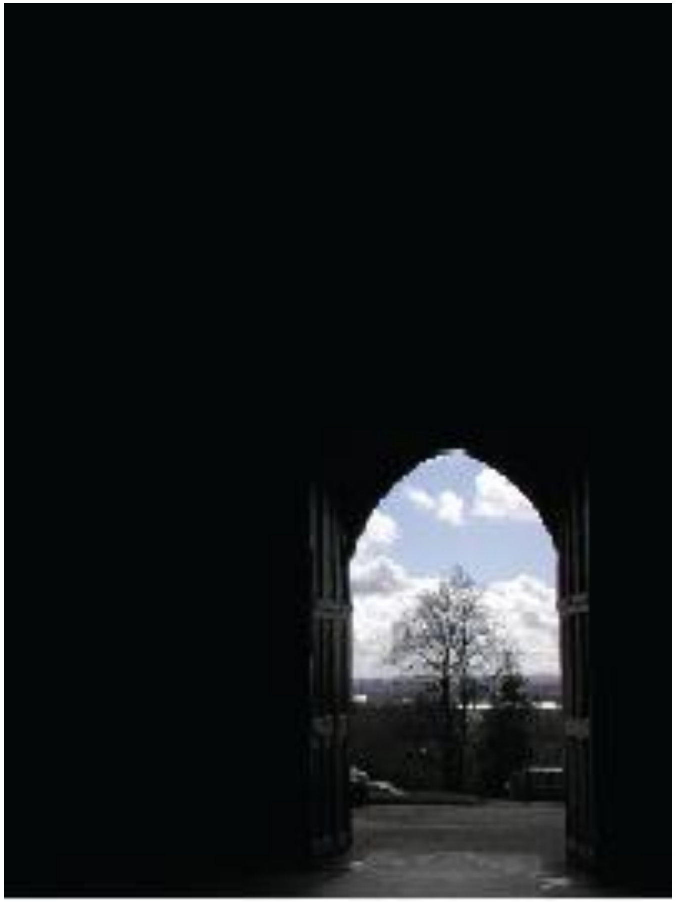
- **Question**: How do we record images in a potentially immortal format?
  - Impossible to predict future technology
  - Reasonable to assume that human visual system will remain as is
- *Dynamic range* of an image: The ratio of its highest to its lowest intensity value
- Human eye has tremendous dynamic range capabilities (10.000:1)
- Conventional displays' typical dynamic range is 300:1
- Conventional 24-bit RGB encoding has a dynamic range of 90:1
  - 24-bit RGB encoding does a relatively good job of representing what a monitor can display
  - 24-bit RGB encoding does a very poor job of representing what the human eye can perceive
  - Dynamic range of conventional camera film is higher than that of 24-bit RGB

High Dynamic Range Images (2)

- HDR images can be produced:
  - By specialized photography equipment
  - By combining multiple images of a scene taken at different brightness levels
  - Synthetically (Global illumination techniques)
- *Tone-mapping* methods: Compress HDR images into the dynamic ranges of monitors according to specific preservation *intents*
- Missing is the capability to display a wide dynamic range simultaneously (e.g. oncoming traffic at night)
- There are two advantages to creating HDR images:
  - Images can be saved for posterity at the dynamic range perceivable by humans
  - Possible to apply *different* tone-mapping techniques to HDR images

High Dynamic Range Images (3)

- Images of a scene with high dynamic range:
  a) A dark image loses information on the interior of the arch
  b) A bright image loses information on the clouds



High Dynamic Range Images (4)

- Images of a scene with high dynamic range:
  - c) HDR image created from several simple images & tone mapped using histogram tone mapping
  - d) Reinhard's global photographic tone mapping, is closer to what the human eye can see

High Dynamic Range Images (5)

- Possible to record HDR images by increasing the bits per pixel
  - E.g. 32 bits per color component for a total of 96 bpp
- HDR formats make clever use of the notion of Just Noticeable Difference (JND)
- JND is the smallest intensity difference detectable by the human eye at a given intensity level
- Logarithmic relationship between JNDs and intensity levels:
  - It makes sense to separate the intensity component of a pixel from its chromatic content and store it separately, encoded at a logarithmic scale
- The above approach is followed by HDR formats, such as RGBE of Radiance & LogLuv

High Dynamic Range Images (6)

- Focus on 32-bit LogLuv:
  - 32 bpp
  - 15 bits for the intensity value
  - 1 bit for the intensity sign (negative intensity is allowed)
  - 16 bits for chromaticity
- Logarithmic conversion between real intensity value L and its (integer) stored value L_e is of the form:

$$L_e = \lfloor c_1(\log_2 L + c_2) \rfloor,$$

$$L = 2^{[L_e/c_1 - c_2]}$$

- The above encompasses the full range of perceivable intensity in imperceptible steps

High Dynamic Range Images (7)

- Bit assignments in 32-bit LogLuv: