Chapter 1 - Introduction

computer graphics - any use of computers to create or manipulate images

1. Graphics Areas

modeling - mathematical specification of shape and appearance properties

rendering - creation of shaded images from 3D computer models

animation - illusion of motion through sequences of images

Related areas:

- user interaction
- virtual reality
- visualisation
- image processing
- 3D scanning
- computational photography

2. Major Applications

- video games
- cartoons
- visual effects
- animated films
- CAD/CAM (computer aided design/manufacturing)
- simulation
- medical imaging
- information visualisation

3. Graphics APIs

API (application program interface) - collection of function performing a set of related operations

graphics API - performs basic operations such as drawing images and 3D surfaces into windows on the screen

graphics program - 2 related APIs:

- graphics API for visual output
- user-interface API for input

4. Graphics Pipeline

- software/hardware subsystem efficiently drawing 3D primitives
- optimised for processing 3D triangles with shared vertices
- basic operations map the 3D vertex location to 2D screen position and shade the triangle
- use z-buffer, special memory buffer to brute-force problems
- 4th homogeneous (perspective) coordinate => 4x4 matrices and 4-vectors

5. Numerical Issues

IEEE floating-point:

1. Infinity (∞)
2. Minus infinity ($-\infty$)
3. Not a number (NaN)

+/- 0:

$+a / +\infty = +0$

$+a / -\infty = -0$

$-a / + \infty = -0$

$-a / -\infty = +0$

other rules:

$\infty + \infty = +\infty$

$\infty - \infty = \text{NaN}$

$\infty \times \infty = \infty$

$\infty / \infty = \text{NaN}$

$\infty / a = \infty$

$\infty / 0 = \infty$

$0 / 0 = \text{NaN}$

most importantly:

$+a / +0 = + \infty$

$-a / +0 = \infty$

6. Efficiency

memory access patterns > operations count

1. Straightforward code (intermediate results on the fly rather than stored)
2. Compile in optimised mode
3. Profiling tools for critical bottlenecks
4. Data structures for locality
5. Access assembly code for numeric computation bottlenecks

7. Graphics Programs

7.1. Class Design

vector2 - 2D vector (x,y), stored in length-2 array

vector3 - 3D vector (x,y,z), analogous to vector2

hvector - homogeneous vector with 4 components

-> (vector) addition, subtraction, dot and cross product, scalar multiplication and division

rgb - colour that stores 3 components

-> (RGB) addition, subtraction, multiplication, scalar multiplication and division

transform - 4x4 matrix for transformations

-> matrix multiply + member functions for locations, directions and surface normal vectors

image - 2D arrays of RGB pixels with output operation

optional classes: unit-length vectors, intervals, orthonormal bases, coordinate frames

7.2. Float vs. Double

7.3. Debugging

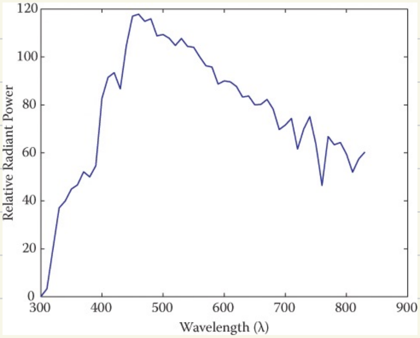
7.3.1. Scientific Method - examine output and hypothesise

7.3.2. Images - output values

7.3.3. Debugger

7.3.4. Data Visualisation

Chapter 19 - Colour



photon - carrier of optical information
propagating through media -> waves
interacting with surface boundary matter -> particles
retina - optical information to electrical signal

$\lambda \Delta E = 1239.9$
λ - wavelength
ΔE - amount of energy (eV)
spectrum - number of photons (energy) plotted against wavelength

colour - aspect of visual perception by which an observer may distinguish differences between two structure-free fields of view of the same size and shape, such as may be caused by differences in the spectral composition of the radiant energy concerned in the observation

1. Colorimetry
    - the science of colour measurement and description
    photodetectors in human retina - rods and cones
    rods - highly sensitive, low-light conditions
    cones - normal lighting conditions,3 types

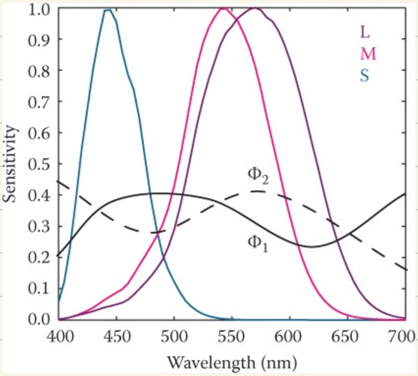
    1.1. Grassmann's Laws
    trichromatic generalisation - any colour stimulus can be matched completely with an additive mixture of three appropriately modulated colour sources
    Symmetry law: colour stimulus A matches stimulus B, then B matches A
    Transitive law: A matches B and B matches C, then A matches C
    Proportionality law: A matches B, then aA matches aB (a - pos. scalar)
    Additivity law: A matches B, C matches D, A + C matches B + D, then A + D matches B + C

1.2. Cone Responses

each cone type - sensitive to a range of wavelengths

sensitivity to wavelength - has peak wavelength with greatest sensitivity

Φ(λ) - stimulus' spectral composition



response - tristimulus values

$L = \int \Phi(\lambda)L(\lambda)\, d\lambda$

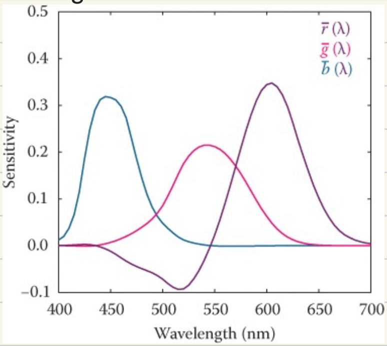
$M = \int \Phi(\lambda)M(\lambda)\, d\lambda$

$S = \int \Phi(\lambda)S(\lambda)\, d\lambda$

1.3. Colour Matching Experiments

metamerism - two spectral different compositions $\Phi1(\lambda)$, $\Phi2(\lambda)$ yield the same response (L, M, S) after integration

- key feature allowing construction of colour reproducing devices



1.4. Standard Observers

three standard primaries - monochromatic light sources of 435.8 (B), 546.1(G), 700 (R) nm

1.5. Chromaticity Coordinates

three primaries = three axes colour space

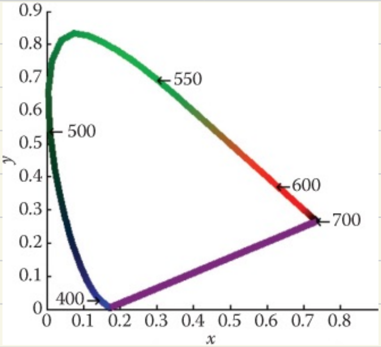
spatial volume = colour gamut

y-axis = luminance => 2D chromatic information

$X = (x/y)\ Y$

$Z = (1-x-y)/y * Y$

chromatic locus ->



Chapter 4 - Ray Tracing

rendering -> making 2D image of 3D objects

object-order - consider all pixels each object influences

image-order - consider all objects each pixel is influenced from (ray-tracing)

1. Basic Ray-Tracing Algorithm
    1) ray generation - computes origin and direction of each pixel's viewing ray
    2) ray intersection - finds closest object intersecting the viewing ray
    3) shading - computes pixel colour based on ray intersection

    example of 2 -> only T2 is printed